

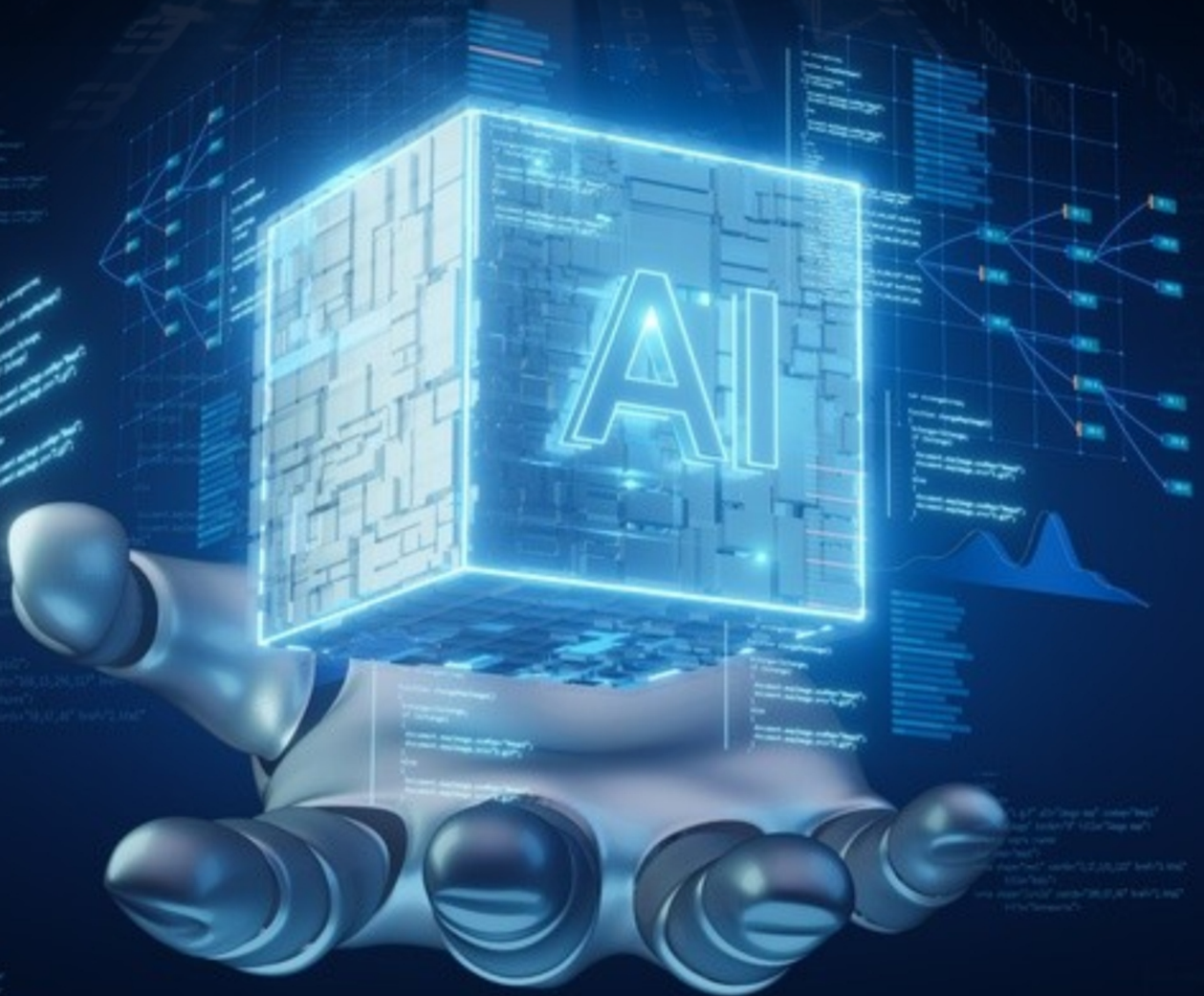


YAYASAN PRIMA AGUS TEKNIK



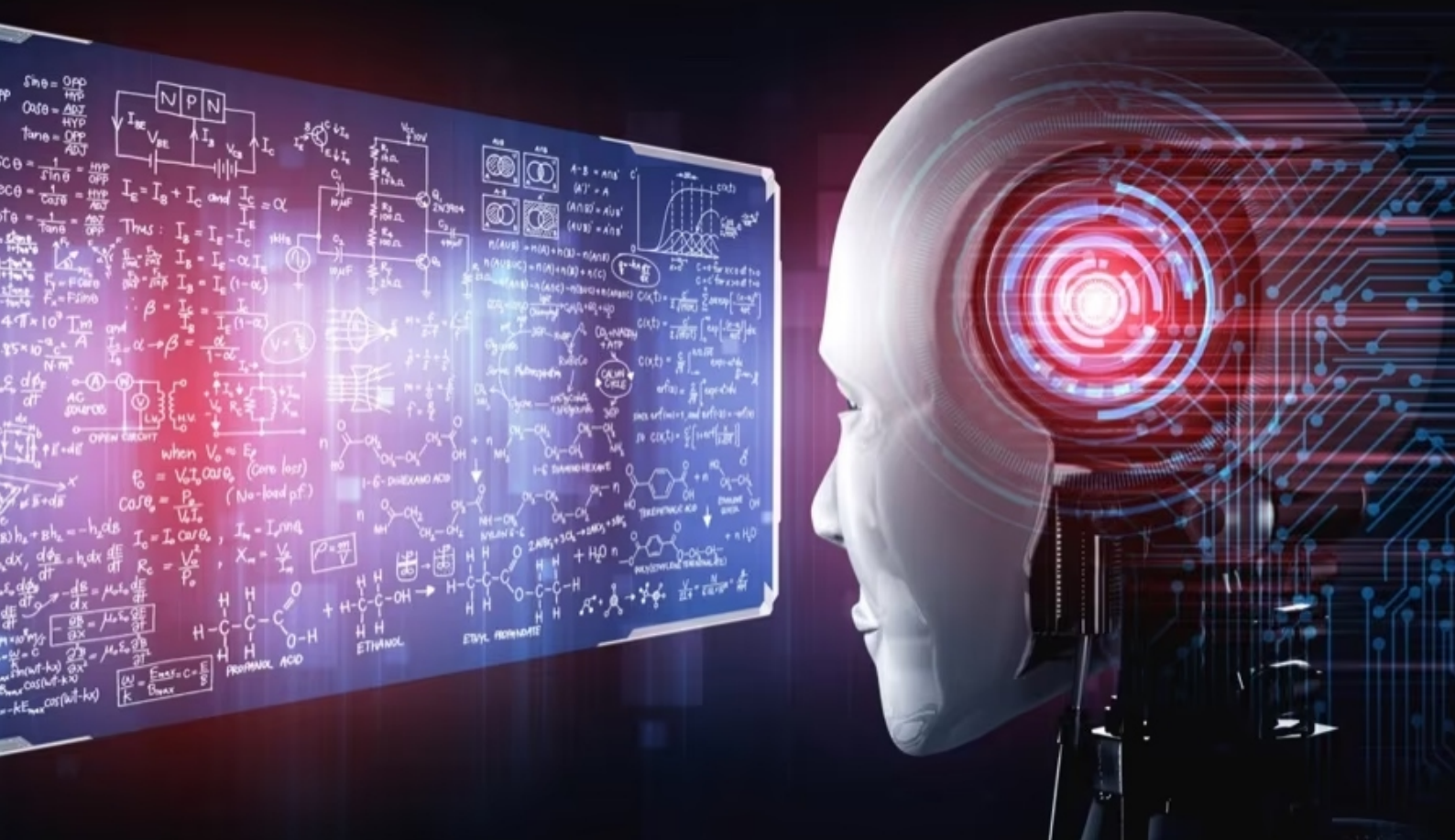
ALGORITMA CERDAS

Untuk Aplikasi AI, IoT dan Otomatisasi



Dr. Joseph Teguh Santoso, S.Kom, M.Kom.

Dr. Joseph Teguh Santoso, S.Kom, M.Kom.



ALGORITMA CERDAS

Untuk Aplikasi AI, IoT dan Otomatisasi



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7695-41-3 (PDF)



9

786347

695413

ALGORITMA CERDAS Untuk Aplikasi AI, IoT dan Otomatisasi

Penulis :

Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

ISBN : 978-634-7695-41-3 (PDF)

Editor :

Dr. Agus Wibowo, M.Kom, M.Si, MM.

Penyunting :

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :

Irdha Yuniarto, S.Ds., M.Kom

Penebit :

Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :

Jl. Majapahit no 605 Semarang

Telp. 08122925000

Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :

Universitas STEKOM

Jl. Majapahit no 605 Semarang

Telp. 08122925000

Fax. 024-6710144

Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin dari penulis

KATA PENGANTAR

Dengan penuh rasa syukur dan antusiasme saya persembahkan buku ini, "Algoritma Cerdas untuk Aplikasi AI, IoT dan Otomatisasi", sebagai panduan komprehensif bagi praktisi, peneliti, dan mahasiswa yang ingin memahami dan menerapkan algoritma modern dalam lingkungan cerdas.

Perkembangan pesat teknologi kecerdasan buatan (AI), Internet of Things (IoT), dan otomatisasi telah mengubah lanskap ilmu pengetahuan dan praktik teknik secara fundamental. Buku ini lahir dari kebutuhan untuk menjembatani gagasan teoretis tentang kompleksitas dan algoritma dengan aplikasi nyata yang menyentuh berbagai bidang kehidupan — dari layanan kesehatan dan pertanian hingga transportasi, energi, dan pemerintahan. Kami menyusun materi ini agar pembaca memperoleh landasan konseptual yang solid, alat-alat praktis untuk pengembangan sistem cerdas, serta wawasan etis dan kebijakan yang diperlukan untuk penerapan yang bertanggung jawab.

Struktur buku dirancang bertahap: bagian awal memperkenalkan konsep kompleksitas — bagaimana sistem kompleks berbeda dari sistem sederhana, metrik efisiensi yang relevan, dan berbagai jenis kompleksitas yang memengaruhi desain algoritma. Bab-bab tentang kompleksitas komputasi, komunikasi, dan data menjelaskan batasan-batasan teoretis serta teknik pengukuran yang membantu kita menilai performa dan kebutuhan sumber daya algoritma dalam konteks nyata.

Selanjutnya, buku ini menguraikan pengukuran risiko, etika, dan batasan algoritma, topik yang menjadi sangat penting ketika sistem otomatis berinteraksi langsung dengan masyarakat. Pembaca akan menemukan pembahasan tentang bias algoritma, keadilan, tata kelola, serta praktik terbaik untuk mengurangi risiko dan meningkatkan efisiensi hijau—semua disajikan dengan contoh dan latihan yang memudahkan pemahaman.

Bagian inti teknis menyajikan paradigma dan fondasi algoritma cerdas, diikuti oleh bab-bab mendetail tentang optimasi, pengambilan keputusan, prediksi, dan keamanan. Di sini pembaca akan mempelajari metode optimasi klasik dan modern, teknik pembelajaran penguatan dan prediksi statistik, serta prinsip-prinsip kriptografi hemat daya yang penting untuk perangkat IoT. Bab latihan pada setiap bagian memberikan kesempatan untuk menerapkan teori dalam masalah nyata dan memperdalam keterampilan analitis.

Buku ini juga menonjolkan aplikasi dunia nyata: pendidikan cerdas, algoritma medis, ekologi dan manajemen energi, pertanian presisi, transportasi cerdas, teknologi informasi untuk masyarakat, pemerintahan cerdas, serta manajemen bencana. Setiap bab aplikasi menjembatani algoritma dengan arsitektur sistem, tantangan domain, dan studi kasus praktis supaya pembaca dapat melihat bagaimana konsep algoritmik diimplementasikan, diukur, dan dievaluasi di lapangan.

Kami harap buku ini tidak hanya menjadi sumber pengetahuan teknis, tetapi juga pendorong diskusi mengenai tanggung jawab sosial dan etika dalam pengembangan teknologi.

Di bagian akhir buku, pembaca akan menemukan rangkaian referensi dan latihan yang dapat digunakan sebagai landasan penelitian lebih lanjut atau sebagai materi pengajaran.

Ucapan terima kasih saya sampaikan kepada para kolega, penelaah, praktisi, dan mahasiswa yang telah memberikan masukan berharga selama penyusunan buku ini. Terima kasih pula kepada institusi dan rekan-rekan yang mendukung akses data dan studi kasus yang memungkinkan bahasan ini bersifat aplikatif. Segala kekurangan menjadi tanggung jawab penulis; kritik dan saran konstruktif sangat kami harapkan untuk penyempurnaan edisi berikutnya.

Selamat membaca. Semoga buku ini memberi Anda pemahaman yang lebih dalam, inspirasi untuk inovasi, dan panduan praktis untuk merancang sistem cerdas yang efektif, aman, dan bertanggung jawab.

Semarang, Juli 2026

Penulis

Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	iii
BAB 1 PENGANTAR KOMPLEKSITAS	1
1.1 Pendahuluan.....	1
1.2 Sistem Kompleks Dan Algoritma	2
1.3 Sistem Kompleks.....	3
1.4 Metrik Efisiensi Untuk Sistem Kompleks	6
1.5 Aplikasi Kompleksitas	9
1.6 Jenis Kompleksitas	13
1.7 Latihan	14
BAB 2 KOMPLEKSITAS KOMPUTASI.....	16
2.1 Komputabilitas.....	16
2.2 Model Komputasi	18
2.3 Kelas Kompleksitas	19
2.4 Kompleksitas Probabilistik.....	21
2.5 Kompleksitas Kuantum	23
2.6 Latihan	26
BAB 3 KOMPLEKSITAS KOMUNIKASI.....	28
3.1 Dasar-Dasar Kompleksitas Komunikasi.....	28
3.2 Komunikasi Deterministik.....	29
3.3 Kompleksitas Komunikasi Deterministik	29
3.4 Komunikasi Nondeterministik	31
3.5 Komunikasi Nondeterministik	31
3.6 Kompleksitas Komunikasi Acak.....	33
3.7 Latihan	35
BAB 4 KOMPLEKSITAS DATA	36
4.1 Teori Informasi Algoritma	36
4.2 Prinsip Occam Dan Inferensi Induktif	38
4.3 Filsafat Informasi	39
4.4 Pelajaran Untuk Filsafat Informasi.....	40
4.5 Metrik Keacakan Kolmogorov.....	41
4.6 Metrik Kompleksitas VC.....	43
4.7 Kompleksitas Rademacher	44
4.8 Latihan	47
BAB 5 PENGUKURAN RESIKO	49
5.1 Mengatasi Bias Algoritma.....	49
5.2 Pengukuran Risiko	50
5.3 Ukuran Keadilan Algoritma.....	51

5.4	Risiko Dalam Monokultur Algoritma	52
5.5	Efisiensi Hijau.....	55
5.6	Kesimpulan	56
5.7	Latihan	57
BAB 6	ETIKA DAN BATASAN ALGORITMA.....	58
6.1	Tata Kelola Algoritma.....	58
6.2	Pengambilan Keputusan Algoritma	59
6.3	Moralitas Algoritma	61
6.4	Etika Sebagai Layanan.....	63
6.5	Tren, Tantangan, Dan Masa Depan.....	64
6.6	Kesimpulan	65
6.7	Latihan	65
BAB 7	PENGANTAR PARADIGMA ALGORITMA CERDAS	68
7.1	Pengantar Paradigma Cerdas	68
7.2	Fondasi Algoritma Cerdas.....	69
7.3	Peta Jalan Untuk Kemajuan Masa Depan.....	70
7.4	Kesimpulan	71
BAB 8	ALGORITMA OPTIMASI.....	72
8.1	Optimasi Dengan Keterbatasan	72
8.2	Metode Minimum Global	73
8.3	Menyelesaikan Persamaan Linear	74
8.4	Dualitas Pemrograman Linier	77
8.5	Masalah Jaringan	83
8.6	Latihan	94
BAB 9	ALGORITMA PENGAMBILAN KEPUTUSAN	96
9.1	Proses Keputusan Markov	96
9.2	Pembelajaran Penguatan.....	108
9.3	Iterasi Nilai.....	109
9.4	Pembelajaran-Q.....	110
9.5	Pembelajaran TD	111
9.6	Latihan	112
BAB 10	ALGORITMA PREDIKSI	114
10.1	Regresi	114
10.2	Klasifikasi Ubah Data	122
10.3	Kompleksitas Model	141
10.4	Algoritma Bayesian.....	149
10.5	Jaringan Saraf	153
10.6	Mesin Vektor Pendukung	155
10.7	Analisis Kluster.....	156
10.8	Model Grafis	162
10.9	Latihan	168

BAB 11 ALGORITMA KRIPTOGRAFI YANG AMAN	170
11.1 Kriptografi Hemat Daya	170
11.2 Kriptografi Kunci Rahasia	171
11.3 Kriptografi Kunci Publik	171
11.4 Latihan	183
BAB 12 PENGANTAR APLIKASI DUNIA CERDAS	184
12.1 Aplikasi Menarik	184
BAB 13 PENDIDIKAN CERDAS.....	189
13.1 Contoh Alat Pendidikan Cerdas	189
13.2 Pembelajaran Yang Dipersonalisasi	190
13.3 Penyampaian Konten Cerdas.....	191
13.4 Analisis Dan Wawasan Pembelajaran.....	192
13.5 Analitik Data Dalam Pendidikan	192
13.6 Tutor Dan Asisten AI Dalam Pendidikan	193
13.7 Evaluasi Pendidikan Terpadu	194
13.8 Pembelajaran Kolaboratif	194
13.9 Latihan	195
BAB 14 ALGORITMA MEDIS CERDAS	197
14.1 Penjadwalan Kapasitas Pelayanan	197
14.2 Teknologi Pengemasan Obat	199
14.3 Keamanan Data Pasien	200
14.4 Sistem Pemantauan Nutrisi Otomatis	201
14.5 Latihan	201
BAB 15 ALGORITMA EKOLOGI DAN ENERGI.....	203
15.1 Urun Daya Pemantauan Udara Kota.....	203
15.2 Manajemen Permintaan Energi Hijau	204
15.3 Jaringan Cerdas.....	206
15.4 Sistem Manajemen Limbah Cerdas	207
15.5 Pemantauan Drone.....	208
15.6 Latihan	210
BAB 16 PERTANIAN CERDAS	211
16.1 Pertanian Presisi	211
16.2 Pemantauan Kesehatan Tanah	213
16.3 Manajemen Irigasi	214
16.4 Prediksi Hasil Panen.....	215
16.5 Latihan	216
BAB 17 TRANSPORTASI CERDAS.....	218
17.1 Manajemen Lalu Lintas Cerdas.....	218
17.2 Sistem Transportasi Cerdas	220
17.3 Transportasi Publik	221
17.4 Parkir Pintar	222

17.5 Kendaraan Otonom	223
17.6 Pemeliharaan Infrastruktur Terpadu	224
17.7 Kendaraan Listrik Dan Terhubung	225
17.8 Tanggap Darurat	226
17.9 Latihan	226
BAB 18 TEKNOLOGI INFORMASI DAN MASYARAKAT	229
18.1 Transformasi Digital Keamanan Publik	229
18.2 Identifikasi Orang Hilang	229
18.3 Penularan Sosial	230
18.4 Penyebaran Penyakit	230
18.5 Pemantauan Kejahatan	231
18.6 Latihan	231
BAB 19 PEMERINTAHAN CERDAS	233
19.1 Layanan E-Government	233
19.2 Utilitas Cerdas.....	235
19.3 Peningkatan Keamanan Publik	236
19.4 Pemantauan Lingkungan	237
19.5 Latihan	238
BAB 20 MANAJEMEN BENCANA	240
20.1 Pendahuluan.....	240
20.2 Sistem Darurat Pasca-Tambang	242
20.3 Manajemen Informasi Bencana	244
20.4 Algoritma Sensor Kebencanaan	245
20.5 Latihan	246
BAB 21 ALGORITMA KOMUNIKASI	249
21.1 Algoritma Komunikasi Untuk WSN	249
21.2 DTN: Algoritma Simpan-Bawa-Teruskan	250
21.3 Algoritma Komunikasi Hemat Energi LLN.....	251
21.4 Algoritma Kontrol Jaringan SDN	251
21.5 Algoritma Jaringan Peer-To-Peer	252
21.6 Latihan	252
DAFTAR PUSTAKA	254



BAB 1

PENGANTAR KOMPLEKSITAS

1.1 PENDAHULUAN

Bab pembuka ini perlu dipahami sebagai fondasi konseptual sebelum memasuki pembahasan teknis algoritma. Kompleksitas tidak hanya berarti “sulit” atau “besar”, tetapi menunjukkan hubungan antara struktur sistem, jumlah interaksi, kebutuhan sumber daya, serta konsekuensi praktis ketika sistem tersebut diterapkan dalam dunia nyata. Dalam konteks AI, IoT, dan otomasi, kompleksitas muncul karena data terus bertambah, perangkat saling terhubung, keputusan dibuat secara otomatis, dan perilaku sistem dapat berubah akibat umpan balik dari lingkungan. Oleh karena itu, pembaca perlu melihat kompleksitas sebagai alat analitis untuk menilai efisiensi, risiko, ketahanan, dan kelayakan implementasi suatu sistem cerdas.

Kompleksitas suatu algoritma mengacu pada jumlah waktu dan/atau ruang yang dikonsumsi saat memproses input berukuran n . Secara lebih luas, total sumber daya—atau biaya komputasi—yang dibutuhkan oleh suatu algoritma untuk menghasilkan hasil yang diinginkan disebut *kompleksitas komputasi*, yang sering disebut sebagai *kompleksitas* saja. Hal ini dapat dievaluasi dalam hal kompleksitas waktu dan kompleksitas ruang. Dalam konteks sistem cerdas, analisis teoretis kompleksitas algoritma masih dalam tahap pembentukan. Hal ini karena algoritma tersebut memerlukan pertimbangan bentuk kompleksitas tingkat lanjut di luar ukuran tradisional—seperti kompleksitas komunikasi, kompleksitas input/output (I/O), kompleksitas data, dan kendala terukur lainnya.

Bab ini membahas:

- a. signifikansi sistem dan algoritma kompleks,
- b. berbagai metrik yang digunakan untuk menilai efisiensi,
- c. tantangan kompleksitas penerapan, dan
- d. dimensi etika dan risiko terkait pengukuran kompleksitas.

Pada akhir bab ini, pembaca akan memperoleh pemahaman komprehensif tentang kompleksitas, yang didukung oleh penjelasan rinci tentang aspek-aspek kunci berikut:

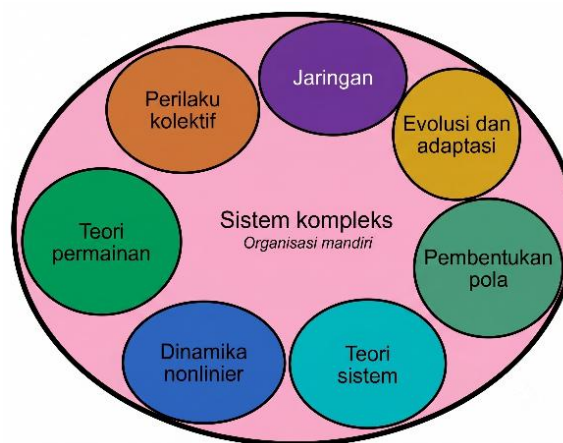
- Memahami berbagai aspek sistem kompleks dan bagaimana algoritma berinteraksi di dalamnya.
- Menyelidiki berbagai metrik untuk mengukur kompleksitas sistem ini.
- Menjelajahi konsep kekerasan dalam kaitannya dengan sistem kompleks dan algoritmanya.

Sistem kompleks adalah sistem di mana setidaknya dua komponen berinteraksi secara dinamis dalam jaringan interkoneksi yang terstruktur secara alami. Bagian-bagian ini saling terkait, dan masing-masing dapat terdiri dari subsistem yang lebih kecil yang tertanam dalam keseluruhan yang lebih besar. Misalnya, seorang individu termasuk dalam sebuah keluarga, yang merupakan bagian dari komunitas, sebuah institusi, sebuah desa, sebuah wilayah,



sebuah negara, sekelompok negara, Bumi, tata surya, galaksi, alam semesta yang dapat diamati, dan pada akhirnya alam semesta itu sendiri.

Sistem kompleks menunjukkan sifat-sifat yang tidak dapat disimpulkan dengan memeriksa elemen-elemen individual secara terpisah. Sistem kompleks umumnya dicirikan oleh beberapa fitur utama: jaringan elemen yang saling terhubung dan saling bergantung; aturan lokal yang mengatur perilaku setiap elemen; pergerakan dan interaksi berkelanjutan antar komponen; kemampuan beradaptasi untuk mempertahankan fungsionalitas dalam kondisi yang berubah; pengorganisasian diri spontan yang menghasilkan pola atau struktur baru; dan kecenderungan untuk berevolusi, menjadi lebih besar dan lebih canggih dari waktu ke waktu (Gambar 1.1). Meskipun tingkat kompleksitas dapat bervariasi di berbagai sistem, semua sistem kompleks menunjukkan kemunculan dan pengorganisasian diri. Selain itu, karakteristik mereka sering berubah seiring waktu secara nonlinier, dan mereka jarang mencapai keseimbangan stabil jangka panjang, meskipun periode stabilitas sementara mungkin terjadi.



Gambar 1.1 Komponen Sistem Kompleks.

1.2 SISTEM KOMPLEKS DAN ALGORITMA

Hubungan antara sistem kompleks dan algoritma bersifat dua arah. Di satu sisi, algoritma digunakan untuk memahami sistem kompleks melalui pemodelan, simulasi, optimasi, dan prediksi. Di sisi lain, algoritma itu sendiri dapat menjadi bagian dari sistem kompleks ketika ia berinteraksi dengan pengguna, data, sensor, jaringan, dan lingkungan operasional. Contohnya, algoritma rekomendasi tidak hanya memprediksi preferensi pengguna, tetapi juga dapat membentuk perilaku pengguna melalui umpan balik berulang. Karena itu, analisis algoritma dalam sistem kompleks harus mempertimbangkan dampak teknis, sosial, dan operasional secara bersamaan. Sistem kompleks tersebar luas di dunia kita, mencakup segala hal mulai dari cara kerja sel yang rumit hingga dinamika kota yang ramai. Sistem ini menentang penjelasan sederhana, menunjukkan sifat-sifat yang muncul dari interaksi rumit berbagai komponen. Memahami sistem kompleks adalah upaya ilmiah yang menarik, dan algoritma memainkan peran penting dalam upaya ini.



1.3 SISTEM KOMPLEKS

Sistem kompleks berbeda dari sistem sederhana karena perilaku totalnya tidak dapat dijelaskan hanya dengan menjumlahkan perilaku setiap komponen. Perubahan pada satu bagian dapat menyebar melalui jaringan interaksi dan menghasilkan efek yang tidak proporsional. Dalam konteks rekayasa, hal ini terlihat pada jaringan listrik pintar, sistem transportasi adaptif, dan ekosistem perangkat IoT. Ketika satu sensor memberikan data keliru, dampaknya dapat memengaruhi modul pemrosesan, keputusan otomatis, dan respons aktuator. Karena itu, pemahaman sistem kompleks membantu perancang sistem mengantisipasi ketidakpastian, dependensi tersembunyi, dan efek domino. Sistem kompleks memiliki beberapa fitur utama:

Fitur Utama Sistem Kompleks

Fitur-fitur tersebut perlu dibaca sebagai karakter yang saling berkaitan. Banyaknya komponen menciptakan ruang interaksi yang luas; interaksi nonlinier menyebabkan respons sistem tidak selalu sebanding dengan penyebabnya; perilaku yang muncul memperlihatkan pola baru yang tidak dirancang secara eksplisit; dan kemampuan adaptasi membuat sistem berubah mengikuti lingkungan. Dalam sistem cerdas, fitur ini terlihat ketika model pembelajaran mesin memperbarui prediksi berdasarkan data baru, ketika kendaraan otonom menyesuaikan rute secara dinamis, atau ketika jaringan sensor mengubah pola komunikasi untuk menghemat energi.

- **Banyak Komponen yang Berinteraksi:** Ini dapat berupa objek fisik, agen, atau bahkan entitas abstrak.
- **Perilaku yang Muncul:** Keseluruhan menjadi lebih besar daripada jumlah bagian-bagiannya, menunjukkan sifat-sifat yang tidak secara langsung dapat dikaitkan dengan komponen individu.
- **Interaksi Nonlinier:** Perubahan kecil dapat memiliki konsekuensi yang signifikan dan tidak dapat diprediksi.
- **Adaptasi dan Evolusi:** Sistem dapat belajar dan menyesuaikan perilakunya dari waktu ke waktu.

Contoh Sistem Kompleks

Contoh biologis, sosial, dan teknologi menunjukkan bahwa kompleksitas bukan fenomena yang terbatas pada komputer. Sistem biologis memiliki regulasi internal, sistem sosial dipengaruhi oleh norma dan keputusan manusia, sedangkan sistem teknologi dibangun dari protokol, perangkat keras, perangkat lunak, dan jaringan. Kesamaan dari ketiganya adalah keberadaan interaksi berlapis. Bagi insinyur komputer dan elektro, analogi lintas-domain ini bermanfaat karena prinsip yang sama—seperti umpan balik, redundansi, adaptasi, dan ketahanan—dapat digunakan untuk merancang sistem teknologi yang lebih stabil.

- **Sistem Biologis:** Dari jaringan pengaturan yang rumit di dalam sel hingga fungsi harmonis suatu ekosistem.
- **Sistem Sosial:** Dinamika interaksi manusia di kota, ekonomi, dan budaya.
- **Sistem Teknologi:** Internet, jaringan listrik, dan jaringan transportasi.



Peran Algoritma Dalam Sistem Kompleks

Algoritma berperan sebagai lensa analitis sekaligus mekanisme intervensi. Sebagai lensa, algoritma membantu menyederhanakan realitas kompleks menjadi model yang dapat dihitung. Sebagai mekanisme intervensi, algoritma mengubah perilaku sistem melalui rekomendasi, kontrol otomatis, alokasi sumber daya, atau prediksi risiko. Tantangannya adalah menjaga agar penyederhanaan tersebut tidak menghapus faktor penting. Model yang terlalu sederhana mungkin cepat dijalankan, tetapi gagal menangkap dinamika sistem; model yang terlalu rumit mungkin lebih akurat, tetapi mahal dan sulit dijelaskan. Algoritma berfungsi sebagai alat yang ampuh untuk mempelajari dan menganalisis sistem kompleks. Berikut beberapa aplikasi utamanya:

Pemodelan Dan Simulasi

Pemodelan dan simulasi memungkinkan pengembang dapat menguji skenario tanpa harus mengganggu sistem nyata. Dalam jaringan listrik, simulasi dapat digunakan untuk melihat dampak beban puncak; dalam transportasi, simulasi dapat memprediksi kemacetan; dalam IoT, simulasi dapat mengevaluasi konsumsi energi dan latensi jaringan. Nilai utamanya terletak pada kemampuan melakukan eksperimen “bagaimana jika” secara aman dan terukur. Namun, kualitas simulasi sangat bergantung pada asumsi model, kualitas data, serta cara interaksi antar komponen direpresentasikan. Algoritma dapat membuat model yang menangkap interaksi penting dalam sistem kompleks, memungkinkan eksplorasi skenario potensial dan prediksi perilaku di masa depan.

Analisis Data

Analisis data pada sistem kompleks tidak hanya mencari pola statistik, tetapi juga mencoba memahami mekanisme yang menghasilkan pola tersebut. Data sensor, log jaringan, transaksi digital, atau interaksi pengguna dapat memperlihatkan tren, anomali, dan hubungan tersembunyi. Dalam praktiknya, analisis data perlu dipadukan dengan pengetahuan domain agar pola yang ditemukan tidak ditafsirkan secara keliru. Korelasi yang kuat, misalnya, belum tentu menunjukkan hubungan sebab-akibat. Karena itu, analisis data harus disertai validasi, interpretasi, dan pengujian terhadap kondisi operasional nyata. Sistem kompleks sering menghasilkan sejumlah besar data. Algoritma dapat digunakan untuk mengidentifikasi pola, mengekstrak wawasan utama, dan memahami mekanisme yang mendasarinya.

Optimasi

Optimasi dalam sistem kompleks jarang memiliki satu tujuan tunggal. Sistem transportasi, misalnya, tidak hanya mengejar waktu tempuh minimum, tetapi juga keselamatan, biaya, konsumsi energi, dan pemerataan layanan. Hal yang sama berlaku pada sistem AI dan otomasi: peningkatan akurasi sering harus diseimbangkan dengan latensi, konsumsi daya, privasi, dan interpretabilitas. Oleh karena itu, optimasi perlu dipandang sebagai proses multiobjektif. Solusi yang “optimal” secara matematis harus tetap dinilai berdasarkan konteks sosial, teknis, dan operasional tempat sistem digunakan. Algoritma dapat digunakan untuk mengoptimalkan kinerja sistem kompleks, yang mengarah pada peningkatan efisiensi, ketahanan, atau hasil yang diinginkan lainnya.



Pembelajaran Mesin

Pembelajaran mesin memperkuat kemampuan sistem kompleks karena model dapat belajar dari data historis dan menyesuaikan prediksi terhadap pola baru. Namun, kemampuan belajar ini juga membawa risiko. Model dapat menyerap bias data, terlalu menyesuaikan noise, atau gagal ketika lingkungan berubah. Dalam sistem dunia cerdas, model pembelajaran mesin sebaiknya tidak hanya dinilai dari akurasi, tetapi juga dari stabilitas, kemampuan generalisasi, keandalan pada data ekstrem, serta dampaknya terhadap keputusan manusia. Evaluasi berkala menjadi bagian penting dari siklus hidup sistem. Teknik seperti jaringan saraf tiruan dapat belajar dari data tentang sistem kompleks, mengungkap hubungan tersembunyi dan memungkinkan pengembangan model prediktif atau strategi kontrol.

Tantangan Dan Peluang Dalam Desain Algoritma

Tantangan utama desain algoritma untuk sistem kompleks adalah menjaga keseimbangan antara ketepatan model dan kelayakan implementasi. Sistem yang melibatkan ribuan sensor, jutaan pengguna, atau jaringan perangkat terdistribusi tidak selalu dapat dianalisis dengan pendekatan deterministik sederhana. Di sinilah peluang muncul: algoritma adaptif, metode aproksimasi, komputasi terdistribusi, dan pembelajaran mesin dapat membantu menangani skala besar. Namun, setiap pendekatan perlu diuji terhadap kegagalan, perubahan data, latensi komunikasi, serta kebutuhan transparansi keputusan.

Meskipun memiliki kekuatan yang besar, algoritma juga menghadapi keterbatasan ketika berurusan dengan sistem kompleks. Jumlah elemen yang berinteraksi dan nonlinieritas yang melekat dapat mempersulit pembuatan model yang akurat dan dapat digeneralisasikan. Selain itu, sistem kompleks sering menunjukkan sifat-sifat yang muncul yang tidak mudah ditangkap oleh pendekatan algoritma tradisional.

Arah Masa Depan

Arah masa depan pengembangan algoritma untuk sistem kompleks akan semakin menekankan kemampuan adaptasi, skalabilitas, dan tanggung jawab. Sistem tidak cukup hanya cepat dan akurat; sistem juga perlu dapat dijelaskan, aman, hemat energi, dan tahan terhadap gangguan. Kombinasi AI, edge computing, digital twin, dan analitik real-time membuka peluang untuk membangun sistem yang dapat memantau dirinya sendiri. Namun, semakin besar otonomi sistem, semakin besar pula kebutuhan tata kelola, validasi, dan audit untuk memastikan keputusan tetap dapat dipertanggungjawabkan.

- Mengembangkan algoritma yang lebih kuat dan terukur yang dapat menangani kompleksitas sistem yang sangat besar ini.
- Menggabungkan konsep dari kecerdasan buatan dan pembelajaran mesin (ML) untuk menciptakan algoritma yang dapat belajar dan beradaptasi bersama dengan sistem kompleks itu sendiri.
- Menjembatani kesenjangan antara teori dan praktik dengan mengembangkan algoritma yang dapat digunakan untuk memecahkan masalah dunia nyata dalam sistem kompleks.

Sistem kompleks menawarkan beragam tantangan dan peluang untuk eksplorasi ilmiah. Algoritma merupakan alat penting dalam upaya ini, menyediakan sarana untuk memodelkan,



menganalisis, dan berpotensi mengendalikan sistem-sistem yang menarik ini. Seiring dengan terus berkembangnya pemahaman dan teknik algoritmik kita, kita dapat mengharapkan wawasan yang lebih dalam tentang cara kerja yang rumit dari dunia kompleks di sekitar kita.

1.4 METRIK EFISIENSI UNTUK SISTEM KOMPLEKS

Efisiensi dalam sistem kompleks perlu diukur secara multidimensi. Ukuran seperti throughput, waktu respons, konsumsi energi, utilisasi memori, dan skalabilitas harus dibaca bersama, bukan secara terpisah. Suatu algoritma dapat memiliki throughput tinggi tetapi boros energi; algoritma lain mungkin hemat memori tetapi menghasilkan latensi tinggi. Dalam aplikasi dunia cerdas, pemilihan metrik harus mengikuti tujuan sistem. Sistem keselamatan kritis lebih menekankan reliabilitas dan batas waktu respons, sedangkan sistem analitik data besar lebih menekankan skalabilitas dan akurasi agregat.

Sistem kompleks, dengan berbagai komponen yang saling bergantung dan perilaku dinamisnya, memerlukan pendekatan yang bernuansa untuk mendefinisikan dan mengukur efisiensi. Metrik efisiensi untuk sistem tersebut harus memperhitungkan berbagai faktor, termasuk throughput, pemanfaatan sumber daya, adaptabilitas, dan skalabilitas. Namun, pemilihan metrik yang tepat bergantung pada tujuan sistem dan konteks di mana sistem tersebut beroperasi.

1. **Efisiensi Throughput:** Salah satu cara mendasar untuk mengevaluasi efisiensi sistem kompleks adalah dengan mengukur throughput—tingkat di mana sistem memproses input untuk menghasilkan output yang diinginkan. Metrik ini umumnya digunakan dalam jaringan produksi dan transportasi, di mana throughput yang lebih tinggi menandakan kinerja yang lebih baik. Misalnya, dalam rantai pasokan, efisiensi throughput dapat mengukur jumlah barang yang berhasil dikirim per satuan waktu.
2. **Efisiensi Pemanfaatan Sumber Daya:** Pemanfaatan sumber daya berfokus pada seberapa efektif suatu sistem menggunakan sumber daya yang tersedia, baik itu daya komputasi, energi, atau bahan baku. Metrik ini sangat relevan dalam sistem di mana meminimalkan pemborosan sangat penting. Sistem yang dapat mencapai tujuannya dengan sumber daya yang lebih sedikit dianggap lebih efisien.
3. **Kemampuan Beradaptasi dan Ketahanan:** Sistem kompleks sering beroperasi di lingkungan yang tidak dapat diprediksi. Kemampuan beradaptasi mengukur kemampuan sistem untuk menyesuaikan perilakunya sebagai respons terhadap perubahan eksternal atau fluktuasi internal. Sistem yang sangat adaptif dapat mempertahankan efisiensi bahkan ketika kondisi menyimpang dari norma. Metrik ini sangat penting dalam mengevaluasi kekokohan sistem seperti jaringan ekologis dan lingkungan komputasi terdistribusi.
4. **Skalabilitas:** Skalabilitas menilai kemampuan sistem untuk mempertahankan atau bahkan meningkatkan efisiensinya seiring bertambahnya ukuran atau kompleksitasnya. Sistem yang skalabel dapat menangani peningkatan permintaan atau kompleksitas tanpa penurunan kinerja. Metrik ini sangat penting dalam menilai sifat tahan masa



depan dari sistem kompleks, seperti platform komputasi awan atau infrastruktur perkotaan.

Tantangan Dalam Mengukur Efisiensi

Tantangan pengukuran efisiensi muncul karena sistem kompleks sering berubah selama pengukuran dilakukan. Data masuk secara terus-menerus, beban kerja tidak stabil, dan interaksi antar komponen dapat menciptakan efek tak terduga. Metrik yang diambil pada kondisi laboratorium belum tentu mewakili performa lapangan. Oleh karena itu, pengukuran efisiensi perlu mencakup uji beban, skenario kegagalan, variasi input, serta evaluasi longitudinal. Dengan cara ini, efisiensi tidak hanya dipahami sebagai performa sesaat, tetapi sebagai kemampuan sistem mempertahankan kinerja dalam kondisi berbeda. Mengukur efisiensi sistem kompleks penuh dengan tantangan karena sifat multidimensionalnya, perilaku yang muncul, dan sifat dinamisnya. Berikut adalah beberapa hambatan utama:

- **Multidimensionalitas:** Sistem kompleks seringkali memiliki banyak tujuan, yang terkadang saling bertentangan. Satu metrik efisiensi mungkin tidak menangkap gambaran lengkap, sehingga diperlukan kombinasi metrik untuk memberikan pandangan yang komprehensif.
- **Perilaku yang Muncul:** Kinerja keseluruhan suatu sistem mungkin tidak secara langsung disebabkan oleh efisiensi komponen individunya. Misalnya, dalam sistem sosial seperti ekonomi, fenomena yang muncul seperti jatuhnya pasar tidak dapat dipahami hanya dengan memeriksa agen individu.
- **Ketersediaan dan Interpretasi Data:** Mengumpulkan dan menginterpretasikan sejumlah besar data yang dihasilkan oleh sistem kompleks dapat menjadi tantangan. Mekanisme pengumpulan dan pemrosesan data yang efisien sangat penting untuk mendapatkan metrik efisiensi yang bermakna.
- **Sifat Dinamis:** Efisiensi sistem kompleks dapat berfluktuasi dari waktu ke waktu karena beradaptasi dengan kondisi yang berubah. Metrik statis tidak cukup; metrik dinamis yang memperhitungkan perubahan yang bergantung pada waktu diperlukan.

Teknik Untuk Mengukur Efisiensi

Teknik pengukuran efisiensi sebaiknya disusun bertingkat. Pada tingkat komponen, pengukuran dapat berfokus pada waktu proses, penggunaan memori, dan konsumsi energi. Pada tingkat jaringan, analisis dapat melihat latensi, throughput, kehilangan paket, dan pola konektivitas. Pada tingkat sistem, model berbasis agen dan pembelajaran mesin dapat digunakan untuk mengevaluasi perilaku kolektif. Kombinasi teknik ini memberikan gambaran lebih lengkap karena satu metode jarang mampu menangkap seluruh dinamika sistem kompleks. Terlepas dari tantangan-tantangan ini, para pengembang telah mengembangkan beragam alat untuk mengukur efisiensi sistem yang kompleks. Teknik-teknik ini meliputi:

1. **Indikator Kinerja:** Melacak aspek-aspek spesifik dari fungsi sistem, seperti kemacetan jaringan atau konsumsi energi per unit output. Indikator-indikator ini dapat memberikan wawasan tentang bagaimana berbagai bagian sistem berkontribusi terhadap efisiensi keseluruhan.



2. **Teknik Analisis Jaringan:** Mengungkap aliran sumber daya dan informasi di dalam sistem, membantu mengidentifikasi hambatan dan inefisiensi. Metode berbasis grafik sangat berguna dalam menganalisis saling ketergantungan dan konektivitas.
3. **Model Berbasis Agen:** Mensimulasikan perilaku komponen individual dan interaksinya, memungkinkan eksplorasi potensi peningkatan efisiensi. Model-model ini efektif dalam menangkap perilaku yang muncul dari interaksi lokal.
4. **Teknik Pembelajaran Mesin (ML):** Mengidentifikasi pola tersembunyi dalam data sistem yang kompleks, yang mengarah pada pengembangan metrik efisiensi berbasis data. Teknik seperti pengelompokan dan analisis komponen utama dapat mengurangi dimensi data, sehingga lebih mudah dianalisis.

Mendefinisikan Efisiensi Dalam Sistem Kompleks: Pendekatan Holistik

Pendekatan holistik berarti tidak memaksakan satu ukuran efisiensi untuk semua sistem. Setiap sistem memiliki tujuan, batasan, dan risiko yang berbeda. Pada sistem medis berbasis AI, efisiensi tidak boleh mengorbankan keselamatan pasien; pada sistem kota cerdas, efisiensi harus mempertimbangkan akses publik dan keberlanjutan; pada sistem industri, efisiensi terkait produktivitas, downtime, dan biaya pemeliharaan. Dengan demikian, metrik harus dirancang sesuai konteks agar hasil evaluasi tidak menyesatkan.

Pendekatan holistik untuk mendefinisikan efisiensi dalam sistem kompleks melibatkan pertimbangan berbagai dimensi dan pemanfaatan teknik analitik canggih. Metrik harus disesuaikan untuk mencerminkan tujuan dan karakteristik unik sistem. Misalnya, metrik yang cocok untuk sistem biologis seperti koloni semut mungkin tidak berlaku untuk sistem sosial seperti jaringan transportasi. Para ahli menganjurkan kombinasi metrik tradisional, seperti throughput dan pemanfaatan sumber daya, bersamaan dengan ukuran yang lebih canggih seperti adaptabilitas dan skalabilitas. Perspektif multifaset ini memastikan bahwa efisiensi ditangkap dari berbagai sudut, memberikan pemahaman yang lebih dalam tentang kinerja sistem.

Integrasi Berbagai Teknik

Integrasi berbagai teknik memungkinkan para pakar dapat melihat sistem dari beberapa sudut. Analisis jaringan menjelaskan struktur hubungan, model berbasis agen menjelaskan perilaku mikro, sedangkan pembelajaran mesin membantu menemukan pola dari data besar. Ketika digabungkan, ketiganya dapat menghasilkan diagnosis sistem yang lebih kuat. Misalnya, dalam jaringan transportasi, analisis jaringan dapat mengidentifikasi simpul kritis, model berbasis agen dapat mensimulasikan perilaku pengendara, dan pembelajaran mesin dapat memprediksi kepadatan lalu lintas berdasarkan data historis. Untuk mengukur efisiensi secara komprehensif, integrasi berbagai teknik seperti pemodelan berbasis agen, analisis jaringan, dan ML direkomendasikan. Pendekatan hibrida ini dapat memberikan wawasan tentang bagaimana berbagai komponen berinteraksi dan berkontribusi pada perilaku sistem secara keseluruhan.

Memanfaatkan Model Dari Ilmu Fisika

Model dari ilmu fisika berguna karena banyak sistem kompleks menunjukkan pola aliran, difusi, stabilitas, dan transisi fase. Konsep seperti entropi, perkolasi, dinamika fluida,



dan teori jaringan dapat diterapkan untuk memahami penyebaran informasi, penyebaran gangguan, atau perubahan struktur sistem. Namun, penggunaan analogi fisika perlu hati-hati. Sistem sosial dan teknologi melibatkan keputusan manusia, aturan institusional, serta desain buatan yang tidak selalu mengikuti hukum alam secara langsung. Model fisika harus dipakai sebagai alat bantu, bukan sebagai penjelasan tunggal. Para pengamat sistem kompleks semakin memanfaatkan model dan matematika dari ilmu fisika untuk mendefinisikan metode sistematis untuk mengukur, memahami, dan memprediksi perilaku makroskopis. Misalnya, model dari hidrodinamika, kimia gas, dan biologi diterapkan untuk memahami aliran informasi dan perilaku yang muncul dalam sistem jaringan skala besar.

Jalan Ke Depan: Menuju Kerangka Kerja Terpadu

Kerangka kerja terpadu untuk kompleksitas perlu menghubungkan ukuran formal, observasi empiris, dan kebutuhan praktis. Ukuran formal memberikan definisi yang ketat; observasi empiris menunjukkan bagaimana sistem berperilaku di lapangan; sedangkan kebutuhan praktis memastikan metrik dapat dipakai oleh pengembang, analis, dan pengambil keputusan. Tanpa integrasi ini, teori kompleksitas berisiko terlalu abstrak, sementara praktik rekayasa berisiko hanya mengandalkan percobaan tanpa dasar konseptual yang jelas.

Pencarian kerangka kerja terpadu untuk mengukur efisiensi dalam sistem kompleks terus berlanjut. Seiring kemajuan kemampuan komputasi dan pemahaman kita tentang interaksi kompleks semakin mendalam, metrik dan teknik analitik baru akan muncul. Merangkul pendekatan multidisiplin yang mengambil dari berbagai bidang seperti fisika, biologi, dan ilmu komputer akan menjadi kunci untuk mengembangkan metrik efisiensi yang kuat yang dapat menangkap esensi sebenarnya dari sistem kompleks. Dengan berfokus pada pendefinisian metrik spesifik konteks dan memanfaatkan perpaduan teknik analitis, kita dapat membuka wawasan baru tentang efisiensi sistem kompleks, membuka jalan untuk mengoptimalkan kinerja dan memastikan keberlanjutan jangka panjangnya.

1.5 APLIKASI KOMPLEKSITAS

Aplikasi kompleksitas menunjukkan bahwa teori ini tidak berhenti pada diskusi abstrak. Kompleksitas digunakan untuk memahami organisasi, ekonomi, pendidikan, pemodelan, jaringan, dan risiko sistemik. Kekuatan pendekatan ini terletak pada kemampuannya membaca keterkaitan antar unsur yang sering tidak tampak jika masalah dipisahkan terlalu jauh. Dalam praktik profesional, cara berpikir kompleksitas membantu insinyur dan manajer menghindari solusi parsial yang memperbaiki satu komponen tetapi menciptakan masalah baru pada tingkat sistem.

Kompleksitas Dalam Praktik

Dalam praktik, pengurangan kompleksitas melalui modularisasi memang berguna, tetapi tidak boleh membuat pengembang mengabaikan antarmuka antar modul. Banyak kegagalan sistem besar terjadi bukan karena modul individual buruk, melainkan karena integrasi, dependensi, dan asumsi antar modul tidak dikelola. Karena itu, manajemen kompleksitas harus mencakup dokumentasi antarmuka, pengujian integrasi, pemantauan



runtime, dan skenario pemulihan. Prinsip ini sangat relevan pada sistem AI-IoT, tempat perangkat, data, jaringan, dan model bekerja secara bersamaan.

Pendekatan tradisional untuk menangani kompleksitas adalah dengan mengurangi atau membatasinya. Biasanya, ini melibatkan kompartementalisasi: membagi sistem besar menjadi bagian-bagian terpisah. Organisasi, misalnya, membagi pekerjaan mereka ke dalam departemen yang masing-masing menangani masalah terpisah. Sistem rekayasa sering dirancang menggunakan komponen modular. Namun, desain modular dapat rentan terhadap kegagalan ketika muncul masalah yang menjembatani divisi tersebut. Meskipun modularitas membantu mengelola kompleksitas dalam komponen individual, interaksi dan ketergantungan antar modul dapat menyebabkan masalah sistemik yang tidak terduga, yang membutuhkan metodologi integrasi dan pengujian yang canggih untuk memastikan ketahanan.

Manajemen Kompleksitas

Manajemen kompleksitas membutuhkan mekanisme koordinasi, bukan hanya pembagian pekerjaan. Dalam proyek besar, setiap keputusan desain dapat memengaruhi biaya, jadwal, keamanan, dan interoperabilitas. Kerangka seperti rekayasa sistem, manajemen risiko, dan tata kelola data dapat membantu organisasi mengelola hubungan antar komponen. Pendekatan adaptif juga penting karena kebutuhan sistem sering berubah selama proyek berjalan. Dengan demikian, manajemen kompleksitas harus menggabungkan disiplin teknis dan fleksibilitas organisasi.

Seiring proyek dan akuisisi tumbuh dalam skala dan kompleksitas, baik perusahaan maupun lembaga pemerintah menghadapi tantangan signifikan dalam mengelolanya secara efektif. Program-program seperti Future Combat Systems Angkatan Darat menggambarkan kerumitan yang terlibat, di mana banyak komponen yang saling bergantung berinteraksi dengan cara yang tidak dapat diprediksi. Seiring sistem berkembang menjadi lebih saling terhubung dan berbasis jaringan, organisasi harus mengadopsi pendekatan manajemen yang canggih—seperti kerangka kerja Cynefin atau praktik rekayasa sistem—untuk menavigasi dan mengendalikan kompleksitas. Pada saat yang sama, pemerintah perlu menetapkan model tata kelola yang mendorong fleksibilitas, ketahanan, dan transparansi untuk mengurangi risiko dan mencapai hasil proyek yang lebih baik.

Ekonomi Kompleksitas

Ekonomi kompleksitas melihat perekonomian sebagai jaringan kemampuan produktif, bukan sekadar kumpulan transaksi. Indeks Kompleksitas Ekonomi menekankan bahwa negara dengan kemampuan menghasilkan produk beragam dan canggih cenderung memiliki kapasitas pengetahuan yang lebih tinggi. Dalam konteks teknologi, gagasan ini relevan karena inovasi tidak hanya bergantung pada satu perusahaan atau satu produk, tetapi pada ekosistem keahlian, infrastruktur, pendidikan, rantai pasok, dan institusi. Kompleksitas ekonomi membantu menjelaskan mengapa beberapa wilayah lebih cepat beradaptasi terhadap perubahan teknologi.

Selama beberapa dekade terakhir, dalam bidang ekonomi kompleksitas yang sedang berkembang, alat prediksi baru telah dikembangkan untuk menjelaskan pertumbuhan ekonomi. Salah satu contoh yang menonjol adalah Indeks Kompleksitas Ekonomi (ECI), yang



dikembangkan oleh fisikawan MIT Cesar A. Hidalgo dan ekonom Harvard Ricardo Hausmann. ECI mengukur intensitas pengetahuan dari struktur produktif suatu negara dengan mempertimbangkan keragaman ekspor dan kompleksitas produk yang diekspor suatu negara. Berdasarkan ECI, Hausmann, Hidalgo, dan tim mereka di Observatorium Kompleksitas Ekonomi telah menghasilkan perkiraan PDB untuk berbagai tahun, termasuk 2020 dan seterusnya. Model-model ini memberikan perspektif baru tentang pembangunan ekonomi dengan menangkap kompleksitas interaksi ekonomi dan menawarkan wawasan tentang potensi masa depan ekonomi nasional.

Kompleksitas Dan Pendidikan

Dalam pendidikan, teori kompleksitas membantu menjelaskan bahwa pembelajaran tidak hanya dipengaruhi oleh materi ajar, tetapi juga oleh interaksi sosial, motivasi, jaringan teman sebaya, teknologi, dan budaya institusi. Hasil belajar muncul dari kombinasi faktor tersebut. Oleh karena itu, pendekatan pendidikan berbasis kompleksitas mendorong penggunaan analisis jaringan, observasi dinamika kelas, dan evaluasi proses pembelajaran secara longitudinal. Perspektif ini berguna untuk merancang intervensi yang lebih kontekstual dan tidak menyederhanakan keberhasilan belajar menjadi satu variabel tunggal.

Teori kompleksitas telah diterapkan pada pengembangan sistem pendidikan, khususnya dalam memahami isu-isu ketekunan siswa dan dinamika pembelajaran. Forsman, Moll, dan Linder mengeksplorasi kelayakan penggunaan ilmu kompleksitas sebagai kerangka kerja untuk memperluas aplikasi metodologis untuk pengembangan sistem pendidikan fisika (PER). Mereka menemukan bahwa membingkai analisis jaringan sosial dalam perspektif ilmu kompleksitas menawarkan penerapan baru dan kuat di berbagai topik PER. Misalnya, pemodelan sistem kompleks telah digunakan untuk memahami bagaimana interaksi antar teman sebaya dan jaringan pembelajaran berkembang dari waktu ke waktu, memberikan wawasan yang mungkin diabaikan oleh metode pengkajian terhadap pendidikan tradisional.

Kompleksitas Dan Pemodelan

Pemodelan sistem kompleks harus dibedakan dari prediksi deterministik sederhana. Model yang baik tidak selalu menghasilkan angka pasti, tetapi dapat mengungkap pola, kecenderungan, sensitivitas, dan kondisi batas. Dalam ekonomi, kesehatan, iklim, atau lalu lintas, model lebih berguna jika membantu pembuat keputusan memahami kemungkinan skenario dan konsekuensi kebijakan. Dengan demikian, nilai model terletak pada kemampuannya memperjelas struktur masalah, bukan hanya pada ketepatan prediksi tunggal.

Friedrich Hayek memberikan kontribusi mendasar pada teori kompleksitas dengan membedakan antara kemampuan kita untuk memprediksi perilaku sistem sederhana dan kapasitas terbatas kita untuk meramalkan dinamika sistem kompleks melalui pemodelan. Ia berpendapat bahwa bidang-bidang seperti ekonomi, bersama dengan disiplin ilmu lain yang melibatkan fenomena kompleks seperti biologi dan psikologi, tidak boleh dimodelkan dengan cara yang sama seperti ilmu pengetahuan yang berurusan dengan sistem sederhana seperti fisika. Menurut Hayek, model sistem kompleks hanya dapat menawarkan pola atau tren, bukan prediksi yang tepat. Teknik modern seperti pemodelan berbasis agen (ABM) membantu



mengatasi tantangan ini dengan mensimulasikan interaksi antar agen individu, sehingga memberikan wawasan tentang perilaku yang muncul yang menjadi ciri sistem kompleks.

Kompleksitas Dan Teori Kekacauan

Teori kekacauan menekankan sensitivitas terhadap kondisi awal. Perubahan kecil pada input dapat menghasilkan perbedaan besar pada output, terutama dalam sistem nonlinier. Hal ini penting bagi desain algoritma karena kesalahan pengukuran kecil, noise sensor, atau data awal yang bias dapat menyebar menjadi keputusan yang sangat berbeda. Dalam sistem keselamatan kritis, prinsip ini menuntut validasi input, toleransi kesalahan, dan analisis sensitivitas. Dengan demikian, teori kekacauan memperingatkan bahwa stabilitas sistem harus diuji, bukan diasumsikan.

Friedrich Hayek memberikan kontribusi mendasar pada teori kompleksitas dengan membedakan antara kemampuan kita untuk memprediksi perilaku sistem sederhana dan kapasitas terbatas kita untuk meramalkan dinamika sistem kompleks melalui pemodelan. Ia berpendapat bahwa bidang-bidang seperti ekonomi, bersama dengan disiplin ilmu lain yang melibatkan fenomena kompleks seperti biologi dan psikologi, tidak boleh dimodelkan dengan cara yang sama seperti ilmu pengetahuan yang berurusan dengan sistem sederhana seperti fisika. Menurut Hayek, model sistem kompleks hanya dapat menawarkan pola atau tren, bukan prediksi yang tepat. Teknik modern seperti pemodelan berbasis agen (ABM) membantu mengatasi tantangan ini dengan mensimulasikan interaksi antar agen individu, sehingga memberikan wawasan tentang perilaku yang muncul yang menjadi ciri sistem kompleks.

Kompleksitas Dan Ilmu Jaringan

Ilmu jaringan memberikan bahasa formal untuk mempelajari hubungan antar komponen. Simpul, sisi, derajat, sentralitas, modularitas, dan jalur terpendek dapat digunakan untuk mengukur struktur sistem. Dalam jaringan listrik, simpul kritis dapat menunjukkan titik rawan gangguan; dalam media sosial, struktur jaringan dapat menjelaskan penyebaran informasi; dalam sistem biologis, jaringan protein membantu memahami fungsi sel. Dengan memetakan struktur jaringan, para pakar dapat mengidentifikasi kerentanan, memperkuat redundansi, dan merancang strategi pemulihan. Sistem kompleks biasanya terdiri dari banyak komponen yang berinteraksi, yang dapat dimodelkan secara efektif menggunakan representasi jaringan—di mana simpul menunjukkan elemen individu dan tautan menangkap interaksi mereka. Misalnya, Internet sering digambarkan sebagai jaringan dengan komputer sebagai simpul dan tautan komunikasi langsung sebagai tepi. Ketahanan sistem tersebut terhadap peristiwa kegagalan telah diperiksa melalui teori perkolasi, sebuah metode dalam analisis sistem kompleks. Sistem dunia nyata lainnya yang dimodelkan sebagai jaringan kompleks meliputi struktur sosial, lembaga keuangan yang saling terkait, sistem transportasi, rute penerbangan, sistem biologis, dan jaringan iklim global. Memahami dinamika kegagalan dan mekanisme pemulihan jaringan ini tetap menjadi bidang pengembangan yang aktif.

Kontributor utama terhadap kerapuhan jaringan adalah kontrol terpusat—di mana kegagalan simpul kritis dapat menyebabkan gangguan yang meluas. Untuk mengatasi hal ini, model berbasis perkolasi menyarankan penggunaan simpul yang diperkuat atau redundan yang mempertahankan koneksi alternatif, sehingga meningkatkan ketahanan jaringan. Ilmu



jaringan telah lebih lanjut digunakan untuk mengeksplorasi kompleksitas sistem Bumi, termasuk interaksi antara ekosistem dan jaringan iklim. Model seperti jaringan "dunia kecil" dan "bebas skala" juga telah menghasilkan wawasan berharga tentang ketahanan, kerentanan, dan dinamika penyebaran dalam sistem yang kompleks.

Kompleksitas Dan Arah Penelitian Masa Depan

Studi tentang kompleksitas terus berkembang, dengan munculnya bidang interdisipliner baru, seperti ilmu jaringan dan ekonomi kompleksitas. Arah pengembangan sistem masa depan meliputi pengembangan kerangka kerja terintegrasi yang menggabungkan teknik dari berbagai domain, meningkatkan pemahaman kita tentang ketahanan sistem, dan menciptakan alat yang lebih efektif untuk mengelola dan memprediksi perilaku dalam sistem adaptif kompleks (CAS). Upaya tersebut akan membuka jalan bagi perancangan sistem yang tangguh dan tahan banting di berbagai domain, dari infrastruktur hingga jaringan sosial-ekonomi.

1.6 JENIS KOMPLEKSITAS

Pembedaan jenis kompleksitas penting karena setiap jenis menyoroti aspek yang berbeda. **Kompleksitas algoritmik** berfokus pada panjang deskripsi; kompleksitas komputasi pada waktu dan ruang; kompleksitas dinamis pada umpan balik dan perubahan; kompleksitas komunikasi pada pertukaran informasi; sedangkan kompleksitas data pada struktur dan keteraturan informasi. Tanpa pembedaan ini, istilah kompleksitas dapat menjadi terlalu luas. Dalam analisis sistem cerdas, pengamat perlu menjelaskan jenis kompleksitas mana yang sedang dibahas agar evaluasi menjadi tepat dan dapat diuji. Akar penyebab kurangnya definisi kompleksitas yang terpadu adalah karena sebenarnya ada beberapa jenis kompleksitas. Perlakuan formal pertama terhadap kompleksitas berfokus pada kompleksitas algoritmik, yang mencerminkan persyaratan komputasi yang dibutuhkan untuk menggambarkan suatu proses matematika. Kompleksitas algoritmik, juga dikenal sebagai kompleksitas Kolmogorov, didefinisikan sebagai panjang deskripsi terpendek yang mungkin (dalam bahasa universal tetap) dari suatu string atau objek tertentu.

Senge dan Stermann juga memasukkan **kompleksitas dinamis**, yang terutama dicirikan oleh hubungan sebab-akibat yang sulit dibedakan karena interaksi nonlinier, lingkaran umpan balik, dan penundaan waktu. Kompleksitas dinamis sering dianalisis menggunakan prinsip-prinsip dari dinamika sistem atau teori kontrol. Ada juga konsep **CAS** (Common Algorithm System). Sistem ini didefinisikan oleh sifat-sifat seperti terdiri dari jaringan agen, memiliki beberapa tingkat organisasi, mengembangkan model internal, menunjukkan 'transisi fase', dan memanfaatkan ceruk dalam lanskap kebugaran. CAS lazim di bidang-bidang seperti biologi, di mana ekosistem beradaptasi dari waktu ke waktu, dan dalam ekonomi, di mana pasar berkembang karena interaksi agen individu.

Salah satu definisi yang paling mudah diterapkan adalah **kedalaman termodinamika**, yang bertujuan untuk menyatukan berbagai kelompok kompleksitas. Kedalaman termodinamika, yang diperkenalkan oleh Seth Lloyd, menyatakan bahwa kompleksitas adalah "ukuran seberapa sulit untuk menyusun sesuatu," yang dikuantifikasi oleh jumlah langkah yang



diperlukan untuk membangun sistem dari komponen-komponennya yang paling sederhana. Kedalaman termodinamika adalah nol untuk sistem yang terurut sempurna dan sistem yang sepenuhnya acak, karena mereka tidak memiliki langkah-langkah kompleksitas perantara.

Bar-Yam mendefinisikan kompleksitas suatu sistem fisik sebagai panjang string terpendek, s , yang dapat mewakili sifat-sifatnya, yang diukur melalui pengamatan dari waktu ke waktu. Definisi ini, yang berasal dari teori informasi, mirip dengan kompleksitas algoritmik tetapi disesuaikan untuk sistem fisik. Namun, ada kesulitan inheren dalam menggunakan definisi ini, khususnya untuk sistem tingkat makro dengan struktur tersembunyi. Meskipun ukuran berbasis informasi dan entropi termodinamika membentuk dasar teoritis untuk pengukuran kompleksitas, masih ada kebutuhan akan pendekatan yang lebih mudah diakses. Ada indikasi bahwa energi berperan dalam mendefinisikan kompleksitas. Sebagai aturan umum, sistem yang umumnya diakui sebagai kompleks cenderung memproses lebih banyak energi daripada sistem yang kurang kompleks. Misalnya, peradaban mengonsumsi lebih banyak energi daripada satu kota. Namun, aliran energi semata bukanlah ukuran kompleksitas, karena satu "pipa" berdiameter sangat besar (sistem yang relatif sederhana) dapat dibangun yang sesuai dengan aliran energi melalui sistem kompleks apa pun. Metrik yang mengatasi masalah ini diusulkan oleh Chaisson, yang mengukur kepadatan laju energi dalam Persamaan (1.1):

$$\Phi_m = \frac{E}{t \times m}$$

di mana E adalah aliran energi melalui suatu sistem, t adalah periode waktu, dan m adalah massa sistem. Hasil Chaisson berkorelasi baik dengan gagasan kompleksitas lainnya, menunjukkan bahwa sistem dengan kepadatan laju energi yang lebih tinggi cenderung menunjukkan tingkat kompleksitas yang lebih tinggi. Metrik ini telah divalidasi di berbagai sistem, dari galaksi dan bintang hingga organisme biologis dan peradaban manusia, memberikan ukuran yang konsisten untuk membandingkan kompleksitas di berbagai domain.

1.7 LATIHAN

1. Definisikan kompleksitas komputasi. Diskusikan pentingnya menganalisis kompleksitas waktu dan ruang dari algoritma. Mengapa penting untuk mempertimbangkan keduanya saat mengevaluasi efisiensi suatu algoritma?
2. Identifikasi dan jelaskan berbagai jenis kompleksitas yang disebutkan dalam konteks algoritma dunia cerdas, seperti kompleksitas komunikasi, kompleksitas I/O, dan kompleksitas data. Berikan contoh masing-masing dan diskusikan bagaimana hal itu memengaruhi kinerja sistem secara keseluruhan.
3. Bandingkan dan kontraskan konsep kompleksitas waktu dan kompleksitas ruang. 3. Bagaimana hubungan antara keduanya, dan pertimbangan apa yang mungkin dipikirkan oleh seorang insinyur perangkat lunak ketika mengoptimalkan algoritma untuk satu jenis kompleksitas dibandingkan jenis kompleksitas lainnya?



4. Diskusikan berbagai metode yang digunakan untuk mengukur efisiensi algoritma. Apa saja metrik kompleksitas tingkat lanjut di luar kompleksitas komputasi tradisional? Bagaimana metrik ini memberikan pemahaman yang lebih baik tentang kinerja algoritma dalam sistem dunia nyata?
5. Pilih sistem dunia nyata (misalnya, jaringan transportasi, sistem keuangan, atau aplikasi berbasis IoT) dan jelaskan bagaimana teori kompleksitas dapat digunakan untuk menganalisis dan mengoptimalkan kinerjanya. Kompleksitas apa saja yang terlibat, dan langkah-langkah apa yang dapat diambil untuk mengatasinya?
6. Identifikasi dan jelaskan tantangan yang dihadapi dalam mengukur kompleksitas algoritma cerdas. Bagaimana aspek-aspek seperti kompleksitas dinamis dan perilaku yang muncul dalam sistem kompleks memengaruhi akurasi dan keandalan pengukuran ini?
7. Diskusikan pertimbangan etis dan potensi risiko yang terkait dengan penerapan sistem kompleks. Bagaimana metrik kompleksitas dapat digunakan untuk menilai dan mengurangi risiko ini?
8. Jelaskan bagaimana berbagai jenis kompleksitas (misalnya, kompleksitas penerapan, kompleksitas komunikasi) memengaruhi desain dan penerapan sistem yang kompleks. Strategi apa yang dapat digunakan untuk mengelola kompleksitas ini secara efektif?



BAB 2

KOMPLEKSITAS KOMPUTASI

Bab ini memperluas pembahasan dari kompleksitas sistem menuju kompleksitas komputasi yang lebih formal. Fokus utamanya adalah bagaimana masalah komputasi diklasifikasikan menurut sumber daya yang diperlukan untuk menyelesaikannya. Dalam rekayasa perangkat lunak dan sistem cerdas, pemahaman ini penting karena algoritma yang tampak benar secara logis belum tentu layak dijalankan pada skala data besar. Kompleksitas komputasi membantu insinyur memperkirakan apakah suatu algoritma dapat dipakai dalam sistem real-time, perangkat tepi dengan memori terbatas, pusat data, atau aplikasi berbasis kecerdasan buatan yang membutuhkan pelatihan dan inferensi berulang.

Setiap komputasi membutuhkan energi, yang secara langsung berhubungan dengan *sumber daya sistem* yang dikonsumsi selama proses tersebut. Dalam komputasi, sumber daya ini disebut sebagai sumber daya sistem, dengan yang paling penting adalah prosesor dan memori—karena komputasi melibatkan eksekusi instruksi dan penyimpanan hasil sementara. Kompleksitas komputasi suatu algoritma, yang sering disebut sebagai *kompleksitasnya* saja, mengukur jumlah sumber daya yang dibutuhkan untuk eksekusi, dengan penekanan utama pada waktu dan penggunaan memori.

Bab ini mengeksplorasi konsep mesin komputasi dan mengevaluasi efisiensinya. Kemudian memperkenalkan prinsip-prinsip analisis amortisasi dan membahas signifikansi relasi rekurensi dalam menganalisis algoritma. Setelah membaca ini, pembaca akan memperoleh pengetahuan yang jelas tentang kompleksitas komputasi, termasuk demonstrasi rinci tentang hal-hal berikut:

- **Kerangka Kerja Teoretis untuk Komputasi:** Formalisasi masalah komputasi, sifat masalah keputusan, dan diskusi rinci tentang model komputasi seperti mesin Turing deterministik dan nondeterministik.
- **Kelas Kompleksitas dan Hubungannya:** Eksplorasi kelas kompleksitas seperti P (waktu polinomial), NP (waktu polinomial nondeterministik), PSPACE (ruang polinomial), EXP (waktu eksponensial), BPP (waktu polinomial probabilistik dengan kesalahan terbatas), dan BQP (waktu polinomial kuantum dengan kesalahan terbatas), menyoroti hubungan inklusi dan eksklusinya, hierarki yang diketahui, dan pertanyaan terbuka seperti dugaan P vs. NP.
- **Model Komputasi Tingkat Lanjut:** Model komputasi probabilistik dan kuantum dibahas, bersama dengan kemampuan unik dan implikasinya untuk pemecahan masalah, yang menggambarkan beragam lanskap teori kompleksitas.

2.1 KOMPUTABILITAS

Komputabilitas membedakan masalah yang secara prinsip dapat diselesaikan oleh mesin dari masalah yang tidak dapat diselesaikan secara algoritmik. Namun, teori kompleksitas



melangkah lebih jauh dengan menanyakan apakah masalah yang dapat diselesaikan itu juga dapat diselesaikan secara efisien. Perbedaan ini krusial dalam rekayasa. Masalah yang dapat diputuskan secara teoritis mungkin tetap tidak berguna secara praktis jika membutuhkan waktu eksponensial pada ukuran input nyata. Karena itu, komputabilitas menyediakan batas kemungkinan, sedangkan kompleksitas menyediakan batas kelayakan. Teori kompleksitas komputasi adalah cabang ilmu komputer teoretis yang bertujuan untuk mengklasifikasikan dan membandingkan kesulitan praktis dalam memecahkan masalah yang melibatkan struktur kombinatorial terbatas. Misalnya, pertimbangkan pertanyaan-pertanyaan berikut: Diberikan dua bilangan asli n dan m , apakah keduanya relatif prima? Diberikan rumus proposisional ϕ , apakah rumus tersebut memungkinkan penugasan yang memuaskan? Jika catur dimainkan di papan $n \times n$, apakah putih memiliki strategi kemenangan dari posisi awal yang ditentukan?

Meskipun semua masalah ini dapat diputuskan dari perspektif teori komputabilitas klasik, masalah-masalah tersebut sangat berbeda dalam kompleksitas praktisnya (Gambar 2.1). Misalnya, menentukan apakah dua bilangan bulat relatif prima dapat diselesaikan secara efisien menggunakan algoritma Euclid, yang membutuhkan sejumlah langkah yang proporsional dengan $\log(n)$. Sebaliknya, masalah-masalah selanjutnya, seperti kepuasan dan strategi catur umum, hanya dapat diselesaikan menggunakan teknik brute-force yang membutuhkan eksplorasi ruang kemungkinan yang sangat besar secara eksponensial, sehingga jauh lebih intensif secara komputasi.



Gambar 2.1 Teori Kompleksitas dan Komponen Utamanya.

Teori kompleksitas berupaya membedakan secara ketat antara masalah yang dapat dipecahkan secara praktis dan masalah yang tidak, dengan memperkenalkan kriteria formal untuk keterputusan yang layak. Suatu masalah dianggap dapat diputuskan secara layak jika dapat dipecahkan oleh mesin Turing konvensional dalam sejumlah langkah yang dibatasi oleh fungsi polinomial dari ukuran input. Himpunan semua masalah tersebut membentuk kelas kompleksitas **P**, yang mencakup masalah seperti yang melibatkan bilangan prima relatif yang disebutkan sebelumnya.



Sebaliknya, kelas **EXP** berisi masalah, seperti masalah catur umum, yang membutuhkan waktu eksponensial terhadap ukuran input untuk dipecahkan. Masalah kepuasan proposisional terletak di dalam kelas **NP**, yang terdiri dari masalah-masalah yang solusi yang benarnya dapat diverifikasi—jika tidak harus ditemukan—oleh mesin Turing nondeterministik dalam waktu polinomial. Sebuah dugaan sentral dalam ilmu komputer teoretis, yang secara luas dianggap sebagai salah satu yang paling mendalam, adalah bahwa **P** secara ketat terkandung dalam **NP**, yaitu, $P \subsetneq NP$.

Menentukan apakah kelas kompleksitas tersebut berbeda tetap menjadi pertanyaan terbuka utama di bidang ini. Meskipun demikian, bahkan dalam bentuknya saat ini, teori kompleksitas memberikan hubungan yang mendalam dengan logika, matematika, dan disiplin ilmu terkait. Hubungan ini memberikan kontribusi signifikan terhadap penyelidikan mendasar tidak hanya dalam ilmu komputer tetapi juga dalam filsafat matematika dan epistemologi.

2.2 MODEL KOMPUTASI

Model komputasi berfungsi sebagai alat abstraksi agar algoritma dapat dianalisis secara formal. Mesin Turing, mesin Turing nondeterministik, model probabilistik, dan model kuantum masing-masing menyederhanakan aspek tertentu dari komputasi. Abstraksi ini tidak selalu mencerminkan semua detail perangkat keras modern, tetapi memberikan kerangka yang stabil untuk membandingkan algoritma. Dengan menetapkan model, ukuran input, waktu eksekusi, dan penggunaan ruang, analisis kompleksitas menjadi konsisten dan tidak bergantung pada mesin fisik tertentu.

Model referensi komputasi, yang dilambangkan dengan $\mathfrak{M}$, dipilih untuk merepresentasikan algoritma secara formal. Model ini diasumsikan *masuk akal*, artinya secara akurat menangkap biaya komputasi untuk menjalankan algoritma yang dijelaskan secara informal yang digunakan dalam praktik matematika. Model mesin Turing deterministik, $\mathfrak{T}$, secara konvensional diadopsi untuk tujuan ini. Masalah pengambilan keputusan dikodekan sebagai himpunan input yang sesuai untuk mesin. Misalnya, di bawah model Turing $\mathfrak{T}$, semua masalah X direpresentasikan sebagai himpunan string biner terbatas, yaitu, $X \subseteq \{0,1\}^*$. Hal ini dicapai melalui fungsi pengkodean $\ulcorner \cdot \urcorner : X \rightarrow \{0,1\}^*$, yang disesuaikan dengan sifat X . Misalnya, jika $X \subseteq \mathbb{N}$, maka $\ulcorner n \urcorner$ adalah representasi biner dari n ; jika X terdiri dari rumus dalam bahasa formal $\mathcal{L}$, seperti logika proposisional, maka $\ulcorner \phi \urcorner$ menunjukkan pengkodean biner dari ϕ . Dengan demikian, masalah diidentifikasi dengan citra terkodenya dalam $\{0,1\}^*$, yang umumnya disebut sebagai *bahasa*.

Suatu mesin M dikatakan memutuskan suatu bahasa X jika ia menghitung fungsi karakteristiknya sesuai dengan konvensi model $\mathfrak{M}$. Misalnya, mesin Turing T memutuskan X jika, untuk setiap input $x \in \{0,1\}^*$, ia berhenti dalam keadaan menerima ketika $x \in X$ dan dalam keadaan menolak ketika $x \notin X$. Dalam masalah fungsi, di mana tujuannya adalah untuk menghitung $f: A \rightarrow B$, mesin M dikatakan menyelesaikan f jika $M(x) = f(x)$ untuk semua $x \in A$, di bawah konvensi input-output $\mathfrak{M}$.

Setiap contoh masalah juga memerlukan ukuran input yang terdefinisi dengan baik, yang diwakili oleh sebuah fungsi $|\cdot|: X \rightarrow \mathbb{N}$, yang memastikan kinerja algoritma dapat



dianalisis secara seragam. Untuk input numerik, seperti $X \subseteq \mathbb{N}$, biasanya didefinisikan $|n| = \log_2(n)$, yang sesuai dengan panjang representasi biner. Untuk rumus logika, ukurannya dapat mencerminkan kompleksitas sintaksis, seperti jumlah variabel atau klausa. Dalam masalah graf, contoh input dikodekan sebagai graf $G = \langle V, E \rangle$ dan $|G|$ biasanya bergantung pada ukuran V dan E . Efisiensi mesin dinilai berdasarkan kompleksitas waktunya, yaitu jumlah langkah dasar $\text{time}_M(x)$ yang dibutuhkan untuk berhenti dan menghasilkan hasil untuk input x . Ini diperluas ke sebuah fungsi:

$$t_M(n) = \max\{\text{time}_M(x) : |x| = n\},$$

mewakili kompleksitas waktu kasus terburuk untuk input berukuran n . Demikian pula, kompleksitas ruang $s_M(n)$ adalah jumlah maksimum lokasi memori yang digunakan selama komputasi pada input berukuran n . Mesin dibandingkan berdasarkan pertumbuhan asimtotik kompleksitas waktu dan ruangnya. Diberikan suatu fungsi $f: \mathbb{N} \rightarrow \mathbb{N}$, kelas pertumbuhannya dilambangkan dengan:

$$O(f(n)) = \{g(n) : \exists c, n_0 \forall n \geq n_0, g(n) \leq c \cdot f(n)\}$$

Fungsi-fungsi yang dapat ditangkap dibatasi hingga faktor konstanta. Misalnya, konstanta, $\log(n)$, n , dan fungsi kuadrat seperti $c_1 n^2 + c_2$ termasuk dalam $O(n^2)$, sedangkan fungsi kubik seperti $\frac{1}{1000} n^3$ tidak.

Kita mengatakan mesin M_1 lebih efisien daripada mesin M_2 jika $t_{M_1}(n) \in O(t_{M_2}(n))$ dan sebaliknya tidak berlaku. Perbandingan ini juga berlaku untuk kompleksitas ruang. Suatu masalah X dikatakan memiliki kompleksitas waktu $O(t(n))$ jika mesin penentu yang paling efisien beroperasi dalam batas tersebut. Selain itu, masalah Y dianggap lebih kompleks daripada X jika kompleksitas X secara asimtotik didominasi oleh kompleksitas Y .

2.3 KELAS KOMPLEKSITAS

Kelas kompleksitas membantu mengelompokkan masalah berdasarkan sumber daya yang dibutuhkan. Kelas seperti P, NP, PSPACE, dan EXP bukan sekadar simbol teoretis, melainkan peta konseptual untuk memahami batas efisiensi. Dalam praktik, mengetahui bahwa suatu masalah berada dalam kelas tertentu dapat memandu pemilihan strategi: algoritma eksak, aproksimasi, heuristik, pemrograman dinamis, atau pembatasan input. Karena banyak relasi antar kelas masih menjadi pertanyaan terbuka, teori ini juga menunjukkan batas pengetahuan komputasi modern.

Kelas kompleksitas mengacu pada kumpulan bahasa yang dapat diputuskan dalam batas waktu atau ruang tertentu $t(n)$ atau $s(n)$, relatif terhadap model komputasi tetap. Untuk menghindari kasus patologis—seperti mendefinisikan kelas kompleksitas berdasarkan fungsi non-rekursif—praktik standar adalah membatasi pertimbangan pada fungsi yang dapat dibangun dalam waktu dan ruang. Suatu fungsi $t(n)$ dapat dibangun dalam waktu jika ada



mesin Turing yang berhenti setelah tepat $t(n)$ langkah ketika diberikan input 1^n (untaian unari satu). Demikian pula, suatu fungsi $s(n)$ dapat dikonstruksi ruang jika mesin Turing pada input 1^n mengunjungi tepat $s(n)$ sel pita sebelum berhenti. Fungsi-fungsi yang umum dipelajari seperti $\log(n)$, n^k , 2^n , dan $n!$ memenuhi kondisi konstruktivitas ini.

Untuk komputasi deterministik, kelas kompleksitas biasanya didefinisikan sehubungan dengan model mesin Turing deterministik $\mathfrak{T}$. Diberikan fungsi yang dapat dikonstruksi waktu dan ruang $t(n)$ dan $s(n)$, kelas kompleksitas deterministik berikut didefinisikan sebagai:

$$\mathbf{TIME}(t(n)) = \{X \subseteq \{0,1\}^* \mid \exists T \in \mathfrak{T}, T \text{ decides } X \text{ and runs in time } \leq t(n)\}$$

$$\mathbf{SPACE}(s(n)) = \{X \subseteq \{0,1\}^* \mid \exists T \in \mathfrak{T}, T \text{ decides } X \text{ and uses space } \leq s(n)\}$$

Karena semua polinomial dalam satu variabel dibatasi oleh $O(n^k)$ untuk beberapa k , maka kelas $\mathbf{P}$ dan $\mathbf{PSPACE}$ didefinisikan sebagai:

$$\mathbf{P} = \bigcup_{k \in \mathbb{N}} \mathbf{TIME}(n^k), \quad \mathbf{PSPACE} = \bigcup_{k \in \mathbb{N}} \mathbf{SPACE}(n^k)$$

Kelas standar tambahan meliputi waktu eksponensial dan ruang logaritmik:

Padanan nondeterministik dari kelas-kelas ini didasarkan pada model mesin Turing nondeterministik $\mathfrak{N}$. Ini termasuk:

$$\mathbf{NTIME}(t(n)) = \{X \subseteq \{0,1\}^* \mid \exists N \in \mathfrak{N}, N \text{ decides } X \text{ and runs in time } \leq t(n)\}$$

Oleh karena itu, kelas nondeterministik standar didefinisikan sebagai:

$$\begin{aligned} \mathbf{NP} &= \bigcup_{k \in \mathbb{N}} \mathbf{NTIME}(n^k) \\ \mathbf{NSPACE} &= \bigcup_{k \in \mathbb{N}} \mathbf{NSPACE}(n^k) \\ \mathbf{NEXP} &= \bigcup_{k \in \mathbb{N}} \mathbf{NTIME}(2^{n^k}) \\ \mathbf{NL} &= \mathbf{NSPACE}(\log n) \end{aligned}$$

Banyak hasil fundamental dan masalah terbuka dalam teori kompleksitas berfokus pada relasi inklusi di antara kelas-kelas ini. Serangkaian hasil kunci yang dikenal sebagai *Teorema Hierarki* menunjukkan bahwa kelas-kelas ini diurutkan secara ketat: jika $t_2(n)$ tumbuh secara signifikan lebih cepat daripada $t_1(n)$, maka

$$\mathbf{TIME}(t_1(n)) \subsetneq \mathbf{TIME}(t_2(n))$$

dengan hasil serupa untuk $\mathbf{NTIME}(t(n))$ dan $\mathbf{SPACE}(s(n))$.



2.4 KOMPLEKSITAS PROBABILISTIK

Kompleksitas probabilistik menunjukkan bahwa keacakan dapat menjadi sumber daya komputasi. Algoritma acak sering digunakan ketika solusi deterministik terlalu mahal atau ketika probabilitas kesalahan kecil dapat diterima. Dalam sistem nyata, pendekatan probabilistik banyak muncul pada pengujian primalitas, sampling data besar, optimasi stokastik, hashing, dan algoritma pembelajaran mesin. Keunggulannya adalah efisiensi; tantangannya adalah memastikan probabilitas kesalahan dipahami, dikendalikan, dan sesuai dengan risiko aplikasi.

Kompleksitas probabilistik adalah landasan teori kompleksitas komputasi modern. Teori ini memperkenalkan konsep fundamental algoritma acak, mengeksplorasi kelas kompleksitas BPP, dan membahas aplikasi serta koneksi dengan kelas kompleksitas lainnya. Teori kompleksitas klasik mengklasifikasikan masalah berdasarkan sumber daya komputasi (waktu dan ruang) yang dibutuhkan untuk menyelesaikannya dengan algoritma deterministik. Namun, kompleksitas probabilistik memperkenalkan gagasan keacakan, yang mengarah pada pemahaman yang lebih kaya dan lebih bernuansa tentang kesulitan komputasi. Algoritma acak menggabungkan lemparan koin acak ke dalam komputasinya. Hasil algoritma tidak hanya bergantung pada input tetapi juga pada pilihan acak yang dibuat selama eksekusi. Algoritma acak menawarkan beberapa keuntungan dibandingkan dengan algoritma deterministik. Algoritma acak terkadang dapat mencapai efisiensi yang lebih baik atau menemukan solusi untuk masalah yang terbukti sulit bagi algoritma deterministik.

Kelas Kompleksitas Bpp

BPP menggambarkan masalah yang dapat diselesaikan secara efisien oleh algoritma acak dengan peluang kesalahan yang terbatas. Inti konsep ini adalah keseimbangan antara kecepatan dan keandalan. Jika algoritma dijalankan berulang kali secara independen, probabilitas kesalahan dapat ditekan semakin kecil. Dalam praktik, teknik ini banyak dipakai karena sistem komputasi sering lebih membutuhkan jawaban cepat dengan kepercayaan tinggi daripada bukti deterministik yang mahal. Namun, untuk sistem keselamatan kritis, toleransi kesalahan tetap harus ditetapkan secara ketat. Kelas kompleksitas BPP mencakup semua masalah keputusan yang dapat diselesaikan oleh algoritma acak yang menggunakan waktu polinomial (sebanding dengan ukuran input) dan menghasilkan jawaban yang benar dengan probabilitas tinggi (lebih besar dari $1 - \epsilon$, di mana ϵ adalah konstanta positif kecil). Jaminan probabilitas tinggi memastikan bahwa algoritma tersebut tidak mungkin mengalami kesalahan.

Contoh Kompleksitas Probabilistik

Kompleksitas probabilistik menunjukkan bahwa keacakan dapat menjadi sumber daya komputasi. Algoritma acak sering digunakan ketika solusi deterministik terlalu mahal atau ketika probabilitas kesalahan kecil dapat diterima. Dalam sistem nyata, pendekatan probabilistik banyak muncul pada pengujian primalitas, sampling data besar, optimasi stokastik, hashing, dan algoritma pembelajaran mesin. Keunggulannya adalah efisiensi; tantangannya adalah memastikan probabilitas kesalahan dipahami, dikendalikan, dan sesuai



dengan risiko aplikasi. Algoritma probabilistik telah banyak diaplikasikan di berbagai bidang ilmu komputer, termasuk:

- **Pengujian Bilangan Prima:** Pengujian bilangan prima Miller–Rabin adalah algoritma acak yang secara efisien menentukan apakah suatu bilangan prima dengan probabilitas tinggi.
- **Algoritma Aproksimasi:** Algoritma acak seringkali dapat memberikan solusi aproksimasi yang terbukti baik untuk masalah yang sulit diselesaikan secara tepat dalam waktu polinomial.
- **Algoritma Las Vegas vs. Monte Carlo:** Algoritma acak dapat dikategorikan sebagai algoritma Las Vegas, yang selalu menghasilkan jawaban yang benar tetapi mungkin membutuhkan waktu yang bervariasi, atau algoritma Monte Carlo, yang berjalan dalam waktu polinomial tetapi memiliki probabilitas kesalahan yang kecil. Hubungan antara BPP dan kelas kompleksitas klasik lainnya merupakan area pengembangan yang aktif. Diketahui bahwa $P \subseteq BPP$, yang berarti bahwa setiap masalah yang dapat diselesaikan secara deterministik dalam waktu polinomial, juga dapat diselesaikan oleh algoritma acak dengan probabilitas tinggi dalam waktu polinomial. Namun, pertanyaan apakah $BPP = P$ tetap menjadi masalah yang belum terjawab.

BPP: Komputasi Probabilistik Yang Efisien

BPP dapat dipahami sebagai bentuk komputasi efisien yang memanfaatkan peluang. Jaminan probabilistiknya bukan berarti hasil bersifat sembarangan, melainkan dikontrol melalui batas kesalahan. Prinsip pengulangan independen memungkinkan peningkatan kepercayaan secara eksponensial terhadap jumlah percobaan. Dalam aplikasi AI dan IoT, pemikiran serupa muncul pada estimasi sensor, deteksi anomali, dan sampling data. Selama distribusi kesalahan dipahami, pendekatan probabilistik dapat memberi efisiensi besar tanpa mengorbankan reliabilitas secara berlebihan.

Model komputasi probabilistik, yang memanfaatkan keacakan, sangat penting dalam teori kompleksitas. Tidak seperti mesin Turing nondeterministik, yang tidak praktis untuk algoritma probabilistik, kelas **BPP** ini menyediakan model yang berguna. **BPP** didefinisikan menggunakan mesin Turing probabilistik, yang merupakan mesin Turing standar dengan akses ke generator bit acak. Suatu masalah X berada dalam **BPP** jika terdapat mesin Turing probabilistik C dan konstanta $p > \frac{1}{2}$ sedemikian rupa sehingga:

- C berjalan dalam waktu polinomial untuk semua input.
- Untuk $x \in X$, C menerima dengan probabilitas setidaknya p .
- Untuk $x \notin X$, C menolak dengan probabilitas setidaknya p .

Ini menangkap gagasan algoritma yang membuat pilihan acak tetapi benar dengan probabilitas tinggi. Dengan mengulangi algoritma tersebut, kita dapat mencapai kepercayaan yang sangat tinggi pada hasilnya. Karena mesin Turing deterministik adalah kasus khusus dari mesin Turing probabilistik, $P \subseteq BPP$. Meskipun pernah dianggap bahwa PRIMES mungkin memisahkan kelas-kelas ini, sekarang diketahui bahwa $PRIMES \in P$. Hubungan antara **BPP** dan kelas lain seperti **NP** tetap menjadi pertanyaan terbuka yang signifikan. Meskipun $BPP \subseteq \Sigma_2^P$, tidak diketahui apakah $NP \subseteq BPP$. Apakah keacakan memungkinkan solusi efisien dari masalah



yang tidak dapat dipecahkan secara deterministik masih merupakan area pengembangan yang terbuka.

Arah Masa Depan Dalam Kompleksitas Probabilistik

Kompleksitas probabilistik menunjukkan bahwa keacakan dapat menjadi sumber daya komputasi. Algoritma acak sering digunakan ketika solusi deterministik terlalu mahal atau ketika probabilitas kesalahan kecil dapat diterima. Dalam sistem nyata, pendekatan probabilistik banyak muncul pada pengujian primalitas, sampling data besar, optimasi stokastik, hashing, dan algoritma pembelajaran mesin. Keunggulannya adalah efisiensi; tantangannya adalah memastikan probabilitas kesalahan dipahami, dikendalikan, dan sesuai dengan risiko aplikasi.

Kompleksitas probabilistik adalah area uji coba yang vital, dengan algoritma acak memainkan peran penting dalam ilmu komputer. Memahami hubungan antara **BPP** dan kelas kompleksitas lainnya sangat penting untuk memajukan pengetahuan kita tentang komputasi yang efisien. Penelitian saat ini berfokus pada derandomisasi, yang bertujuan untuk mengubah algoritma acak menjadi algoritma deterministik. Menjelajahi hubungan antara kompleksitas probabilistik dan bidang lain seperti kompleksitas komunikasi dan teori pembelajaran juga memiliki potensi yang signifikan.

2.5 KOMPLEKSITAS KUANTUM

Kompleksitas kuantum memperluas teori komputasi dengan memanfaatkan prinsip mekanika kuantum. Superposisi, interferensi, dan keterikatan memungkinkan model komputasi yang berbeda dari mesin klasik. Namun, keunggulan kuantum tidak berarti semua masalah menjadi mudah. Beberapa algoritma, seperti Shor dan Grover, menunjukkan percepatan penting pada kelas masalah tertentu, tetapi hubungan umum antara BQP, P, dan NP masih belum sepenuhnya diketahui. Karena itu, komputasi kuantum harus dipahami sebagai paradigma yang kuat namun spesifik, bukan pengganti universal komputasi klasik. Bidang komputasi kuantum yang berkembang pesat telah memperkenalkan dimensi baru pada teori kompleksitas komputasi. Kelas kompleksitas BQP mencakup masalah yang dapat dipecahkan oleh komputer kuantum dengan akurasi tinggi dalam waktu polinomial. Namun, batasan pasti BQP dan hubungannya dengan kelas kompleksitas klasik yang sudah mapan seperti P dan NP masih terbuka bagi para pengamat.

Teori kompleksitas komputasi menyelidiki kesulitan inheren dalam memecahkan masalah menggunakan sumber daya komputasi. Kelas kompleksitas klasik seperti P dan NP mengkategorikan masalah berdasarkan kemampuan penyelesaiannya oleh komputer klasik. P mencakup masalah yang dapat diselesaikan secara deterministik dalam waktu polinomial, sedangkan NP mencakup masalah yang verifikasi dapat dilakukan secara efisien, meskipun menemukan solusinya sendiri lebih sulit. Munculnya mekanika kuantum telah merevolusi paradigma komputasi. Komputer kuantum memanfaatkan prinsip superposisi dan keterikatan untuk melakukan komputasi yang tidak dapat dilakukan oleh mesin klasik. BQP mencakup ranah masalah yang dapat diselesaikan oleh komputer kuantum dengan probabilitas kesalahan terbatas dalam waktu polinomial.



BQP: Kekuatan Dan Daya Tarik

Daya tarik BQP berasal dari kemungkinan menyelesaikan masalah tertentu secara jauh lebih cepat daripada algoritma klasik terbaik yang diketahui. Faktorisasi bilangan besar adalah contoh paling terkenal karena berdampak pada kriptografi. Akan tetapi, kekuatan ini bergantung pada kemampuan membangun perangkat kuantum yang stabil, mengendalikan noise, dan melakukan koreksi kesalahan. Dengan kata lain, batas teoretis BQP harus dibedakan dari kesiapan implementasi praktis. Keunggulan teoretis belum otomatis berarti keunggulan operasional pada semua skenario. Kekuatan BQP terletak pada kemampuannya untuk melampaui keterbatasan klasik. Algoritma Shor, misalnya, mencontohkan kehebatan BQP dengan memfaktorkan bilangan bulat besar secara eksponensial lebih cepat daripada algoritma klasik mana pun yang dikenal. Hal ini memiliki implikasi mendalam bagi kriptografi. Namun, jangkauan sebenarnya dari BQP masih belum jelas. Kita tahu bahwa **P** dan **BPP** merupakan himpunan bagian dari **BQP**, yang menyiratkan bahwa setiap masalah yang dapat diselesaikan secara efisien secara klasik juga dapat diselesaikan oleh komputer kuantum.

Namun, kebalikannya belum terbukti. Apakah ada masalah yang dapat diselesaikan oleh komputer kuantum yang secara fundamental sulit dipecahkan oleh mesin klasik? Ini adalah inti dari pertanyaan **P** vs. **BQP**, sebuah tantangan terbuka utama dalam teori kompleksitas komputasi. Namun, kebalikannya masih belum terbukti. Apakah ada masalah yang dapat diselesaikan oleh komputer kuantum yang pada dasarnya tidak dapat dipecahkan oleh mesin klasik? Inilah inti dari pertanyaan **P** vs. **BQP**, sebuah tantangan terbuka utama dalam teori kompleksitas komputasi.

P, NP, Dan BQP

Hubungan antara **P**, **NP**, dan **BQP** menjadi salah satu area penting karena menyangkut batas kemampuan komputer klasik dan kuantum. **P** berisi masalah yang dapat diselesaikan efisien secara klasik, **NP** berisi masalah yang solusinya dapat diverifikasi efisien, sedangkan **BQP** berisi masalah yang dapat diselesaikan efisien oleh komputer kuantum dengan kesalahan terbatas. Pertanyaan kuncinya adalah apakah komputer kuantum dapat menyelesaikan masalah yang tidak dapat diselesaikan secara efisien oleh komputer klasik, dan apakah keunggulan tersebut mencakup masalah **NP**-lengkap.

Memecahkan pertanyaan **P** vs. **BQP** akan memiliki implikasi mendalam bagi lanskap komputasi. Jika **P** = **BQP**, maka komputer kuantum tidak akan menawarkan keunggulan komputasi inheren dibandingkan mesin klasik untuk semua masalah. Sebaliknya, jika **P** ≠ **BQP**, hal itu akan menandakan adanya masalah yang dapat diselesaikan secara efisien hanya pada komputer kuantum, yang mengawali era baru supremasi kuantum. Pertanyaan menarik lainnya menyangkut hubungan antara **BQP** dan **NP**. **NP** mencakup masalah yang solusinya dapat diverifikasi secara efisien, meskipun menemukannya secara langsung mahal secara komputasi. Pertanyaan apakah **BQP** menawarkan prosedur verifikasi yang efisien untuk semua masalah **NP** (yaitu, **BQP** ⊆ **PSPACE**) merupakan batas terbuka lainnya.

Batasan **BQP** dan hubungannya yang rumit dengan kelas kompleksitas klasik tetap menjadi misteri yang memikat. Menyelami lebih dalam lanskap ini memiliki potensi yang sangat besar untuk masa depan komputasi. Menyelesaikan pertanyaan-pertanyaan terbuka ini



tidak hanya akan menerangi kemampuan sebenarnya dari komputer kuantum tetapi juga memandu pengembangan algoritma baru yang ampuh yang memanfaatkan kekuatan unik dari paradigma klasik dan kuantum.

BQP: Komputasi Kuantum Yang Efisien

Komputasi kuantum yang efisien bergantung pada desain algoritma yang mampu mengeksplorasi struktur kuantum. Algoritma Grover memberi percepatan kuadratik untuk pencarian tidak terstruktur, sedangkan Shor memberi percepatan besar untuk faktorisasi. Akan tetapi, tidak semua masalah memperoleh percepatan serupa. Untuk memperoleh manfaat, masalah harus memiliki struktur yang dapat diterjemahkan ke operasi kuantum. Karena itu, uji coba BQP tidak hanya menanyakan apa yang dapat dihitung, tetapi juga masalah mana yang secara alami cocok dengan model kuantum.

Kelas kompleksitas **BQP** (waktu polinomial kuantum kesalahan terbatas) memperluas **BPP** dengan menggunakan komputasi kuantum. Komputasi kuantum memanfaatkan fenomena kuantum seperti keterikatan dan interferensi, beroperasi pada qubit, yang merupakan superposisi dari 0 dan 1. Model seperti Mesin Turing kuantum membentuk dasar **BQP**. Algoritma kuantum, seperti algoritma pencarian Grover dan algoritma faktorisasi Shor, menunjukkan potensi peningkatan kecepatan dibandingkan algoritma klasik. Algoritma Grover mencapai waktu pencarian $O(n^{1/2})$ dibandingkan dengan klasik $O(n)$, dan algoritma Shor memfaktorkan bilangan bulat dalam waktu $O(\log_2(n)^3)$, secara signifikan lebih cepat daripada metode klasik yang diketahui. Karena model kuantum dapat mensimulasikan model klasik, $P \subseteq BPP \subseteq BQP$. **BQP** mengandung masalah seperti FAKTORISASI, yang tidak diketahui berada dalam **P**.

Sementara $BQP \subseteq PSPACE$, hubungannya dengan **NP** tidak jelas. Belum ditemukan algoritma kuantum waktu polinomial untuk masalah **NP**-lengkap pada model kuantum standar. Kelayakan membangun komputer kuantum yang tangguh untuk memecahkan masalah yang secara klasik tidak dapat dipecahkan juga merupakan pertanyaan terbuka utama. Bahkan jika $P \subsetneq BQP$ terbukti, implementasi praktis tetap menjadi tantangan. Meskipun demikian, komputasi kuantum adalah bidang pengembangan yang dinamis.

Arah Masa Depan Dalam Kompleksitas Kuantum

Kompleksitas kuantum memperluas teori komputasi dengan memanfaatkan prinsip mekanika kuantum. Superposisi, interferensi, dan keterikatan memungkinkan model komputasi yang berbeda dari mesin klasik. Namun, keunggulan kuantum tidak berarti semua masalah menjadi mudah. Beberapa algoritma, seperti Shor dan Grover, menunjukkan percepatan penting pada kelas masalah tertentu, tetapi hubungan umum antara BQP, P, dan NP masih belum sepenuhnya diketahui. Karena itu, komputasi kuantum harus dipahami sebagai paradigma yang kuat namun spesifik, bukan pengganti universal komputasi klasik.

Penelitian kompleksitas kuantum berfokus pada pemahaman kemampuan **BQP**. Teknik seperti kueri kuantum dan kompleksitas komunikasi digunakan untuk menganalisis **BQP**. Menjelajahi hubungan antara **BQP** dan kelas seperti **PSPACE** sangat penting. Mengembangkan teori terpadu kompleksitas kuantum, termasuk hubungan **BQP** dengan



kelas klasik, adalah tujuan utama. Penelitian ini bertujuan untuk memberikan pemahaman komprehensif tentang kemungkinan komputasi, membentuk masa depan komputasi.

2.6 LATIHAN

1. Analisis Kompleksitas Komputasi Algoritma: Pilih masalah komputasi yang dapat diselesaikan menggunakan pendekatan deterministik dan nondeterministik. Implementasikan kedua jenis algoritma dan analisis kinerjanya pada berbagai ukuran input. Ukur dan bandingkan kompleksitas waktu dan ruang dari setiap algoritma menggunakan analisis teoretis dan data empiris. Berdasarkan hasil Anda, berikan wawasan tentang perbedaan praktis antara kelas algoritma ini dan diskusikan skenario di mana satu pendekatan mungkin lebih cocok daripada yang lain.
2. Algoritma Probabilistik dan Efisiensinya: Pilih algoritma probabilistik yang terkenal seperti uji prima Miller–Rabin atau quicksort acak. Implementasikan algoritma tersebut dan evaluasi efisiensinya dengan memvariasikan keacakan (misalnya, jumlah percobaan, seed acak) dan ukuran input. Bandingkan kinerjanya dengan versi deterministik dari algoritma yang sama, dengan fokus pada waktu eksekusi, akurasi, dan pemanfaatan sumber daya. Ringkaslah pertimbangan antara metode probabilistik dan deterministik dan diskusikan bagaimana keacakan dapat memengaruhi efisiensi komputasi.
3. Mensimulasikan Algoritma Kuantum untuk Masalah Klasik: Gunakan simulator komputasi kuantum untuk mengimplementasikan algoritma kuantum seperti pencarian Grover atau faktorisasi Shor. Simulasikan eksekusi algoritma dan analisis kinerjanya dibandingkan dengan algoritma klasik yang memecahkan masalah yang sama. Fokus pada kompleksitas waktu dan jumlah operasi yang dibutuhkan dalam setiap kasus. Diskusikan implikasi hasil ini dalam hal keunggulan komputasi yang diberikan algoritma kuantum dibandingkan algoritma klasik, dan identifikasi potensi aplikasi masa depan dari algoritma ini.
4. Menyelidiki Dugaan P vs. NP melalui Perbandingan Eksperimental: Pilih masalah yang diketahui termasuk dalam kelas P dan masalah lain yang NP-lengkap. Implementasikan algoritma untuk kedua masalah tersebut dan lakukan studi eksperimental untuk membandingkan kinerjanya. Analisis bagaimana algoritma berperilaku seiring bertambahnya ukuran input, dan tentukan pada titik mana masalah NP-lengkap menjadi tidak layak untuk dipecahkan dalam waktu polinomial. Sajikan temuan Anda dengan visualisasi dan diskusikan implikasi masalah P vs. NP dalam ilmu komputer teoretis dan aplikasi dunia nyata.
5. Memvisualisasikan Kelas Kompleksitas dan Hubungannya: Buat representasi visual terperinci dari berbagai kelas kompleksitas seperti P, NP, PSPACE, EXP, BPP, dan BQP. Sertakan hubungan inklusi dan eksklusi yang diketahui di antara kelas-kelas ini, dan soroti masalah terbuka penting seperti dugaan P vs. NP. Sertai visual tersebut dengan penjelasan tertulis yang menggambarkan setiap kelas dan posisinya dalam hierarki.



- Diskusikan signifikansi hubungan ini dan bagaimana penyelesaian masalah terbuka dapat memengaruhi pemahaman kita tentang kompleksitas komputasi.
6. Derandomisasi dan Dampaknya pada Kelas Kompleksitas: Selidiki konsep derandomisasi dan perannya dalam mengurangi kompleksitas algoritma probabilistik. Pilih algoritma probabilistik dan jelajahi berbagai teknik derandomisasi untuk mengubahnya menjadi algoritma deterministik. Terapkan teknik-teknik ini dan analisis dampaknya pada kompleksitas waktu dan ruang. Akhiri dengan diskusi tentang potensi manfaat dan keterbatasan derandomisasi dalam kompleksitas komputasi, dan bagaimana hal itu memengaruhi hubungan antara P, BPP, dan kelas kompleksitas lainnya.
 7. Menjelajahi Keunggulan Kuantum Melalui Eksperimen Algoritma: Teliti dan simulasikan algoritma kuantum yang mendemonstrasikan konsep keunggulan kuantum, seperti pengambilan sampel sirkuit acak. Bandingkan eksekusi algoritma ini pada simulator kuantum dengan algoritma klasik. Fokus pada metrik seperti waktu eksekusi, penggunaan memori, dan tingkat kesalahan. Diskusikan kelayakan pencapaian keunggulan kuantum dan tantangan yang dihadapi dalam implementasi praktis. Evaluasi bagaimana kemajuan dalam komputasi kuantum dapat mendefinisikan ulang batasan kelas kompleksitas klasik.
 8. Studi Empiris tentang Teorema Hierarki Waktu: Implementasikan dan analisis dua algoritma yang mendemonstrasikan teorema hierarki waktu. Pilih satu masalah yang dapat diselesaikan dalam waktu polinomial dan masalah lain yang membutuhkan waktu eksponensial. Lakukan eksperimen untuk menunjukkan perbedaan praktis dalam penggunaan sumber daya untuk masalah-masalah ini. Berdasarkan temuan Anda, diskusikan bagaimana teorema hierarki waktu menetapkan hierarki yang tepat di antara kelas kompleksitas dan implikasinya untuk masalah komputasi dunia nyata.



BAB 3

KOMPLEKSITAS KOMUNIKASI

3.1 DASAR-DASAR KOMPLEKSITAS KOMUNIKASI

Bab ini menekankan bahwa biaya komputasi tidak selalu hanya berada pada prosesor atau memori, tetapi juga pada pertukaran informasi antar entitas. Dalam sistem terdistribusi, IoT, jaringan sensor, kendaraan otonom, dan sistem kendali industri, komunikasi sering menjadi hambatan utama karena bandwidth terbatas, latensi tinggi, konsumsi energi meningkat, dan risiko kegagalan jaringan. Kompleksitas komunikasi memberikan kerangka formal untuk menjawab pertanyaan praktis: informasi minimum apa yang harus dikirim agar dua atau lebih pihak dapat mengambil keputusan yang benar? Dengan demikian, bab ini menghubungkan teori kompleksitas dengan desain protokol komunikasi yang hemat, aman, dan dapat diskalakan.

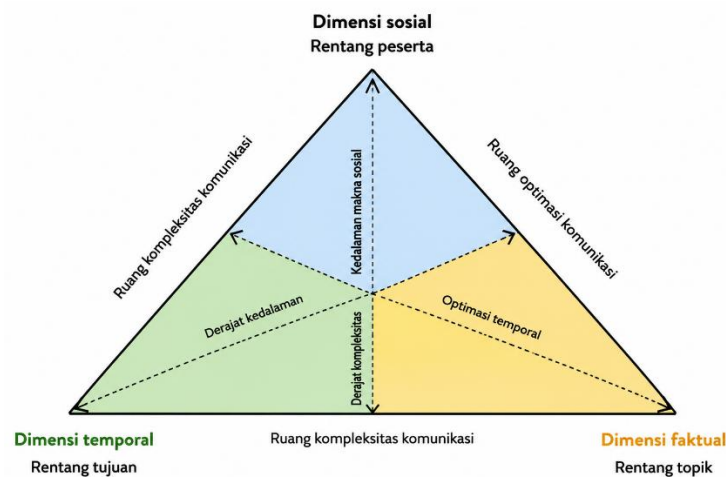
Kompleksitas komunikasi adalah subbidang teori kompleksitas komputasi yang berfokus pada kuantifikasi jumlah komunikasi yang dibutuhkan antara dua pihak atau lebih untuk bersama-sama menghitung suatu fungsi. Dalam model dua pihak klasik, dua pemain—Alice dan Bob—masing-masing diberi sebagian dari input, biasanya dalam bentuk string n -bit x dan y . Tujuan mereka adalah untuk menghitung fungsi $f(x, y)$ yang bergantung pada kedua input sambil bertukar informasi sesedikit mungkin. Model ini diperkenalkan oleh Andrew Yao pada tahun 1979 untuk menganalisis biaya komputasi terdistribusi di beberapa mesin. Sejak itu, model ini telah menemukan berbagai aplikasi dalam kompleksitas sirkuit, struktur data, algoritma streaming, sistem terdistribusi, dan kriptografi. Pertanyaan sentral yang dibahasnya adalah: *Diberikan x dan y yang dipegang oleh Alice dan Bob, berapa banyak bit yang harus mereka komunikasikan untuk menentukan nilai $f(x, y)$?* Kompleksitas komunikasi tidak hanya menangkap biaya inheren pertukaran informasi tetapi juga berfungsi sebagai alat yang ampuh untuk membuktikan batas bawah dalam berbagai model komputasi. Perluasan model, seperti kompleksitas komunikasi deterministik, acak, dan kuantum, memberikan wawasan yang lebih dalam tentang efisiensi dan keterbatasan berbagai paradigma komputasi.

Tujuan utama bab ini adalah untuk memperkenalkan dan mengeksplorasi konsep, model, dan teknik dasar dalam kompleksitas komunikasi (Gambar 3.1). Pada akhir bab ini, pembaca diharapkan dapat:

1. **Memahami Dasar-Dasar Kompleksitas Komunikasi:** Memahami konsep inti, motivasi historis, dan signifikansi kompleksitas komunikasi dalam teori kompleksitas komputasi dan komputasi terdistribusi.
2. **Menganalisis Kompleksitas Komunikasi Deterministik:** Mempelajari model komunikasi deterministik di mana protokol komunikasi menghasilkan hasil yang dapat diprediksi dan tepat, dengan fokus pada meminimalkan jumlah bit yang dipertukarkan antar pihak.



3. **Memeriksa Kompleksitas Komunikasi Nondeterministik:** Mengeksplorasi model nondeterministik di mana protokol dapat menggunakan tebakan atau saksi tambahan, dan mengevaluasi bagaimana fleksibilitas tersebut memengaruhi biaya komunikasi.
4. **Menyelidiki Kompleksitas Komunikasi Acak:** Memahami protokol acak menggunakan teknik probabilistik untuk mengurangi komunikasi, termasuk model dengan keacakan publik dan pribadi serta margin kesalahan yang terkontrol.
5. **Memperoleh Wawasan Teoretis dan Terapan:** Menerapkan alat kompleksitas komunikasi seperti persegi panjang kombinatorial dan teknik berbasis peringkat, dan memeriksa relevansinya melalui bukti formal, contoh masalah, dan aplikasi praktis.
6. **Mengidentifikasi Kompromi dan Keterbatasan Komunikasi:** Mendiskusikan kompromi dan keterbatasan yang melekat dalam berbagai model komunikasi, memungkinkan pilihan yang tepat dalam desain protokol untuk sistem terdistribusi dunia nyata.



Gambar 3.1 Ruang Kompleksitas Komunikasi.

3.2 KOMUNIKASI DETERMINISTIK

Komunikasi deterministik menuntut bahwa protokol selalu menghasilkan keluaran benar untuk setiap pasangan input. Model ini cocok untuk sistem yang tidak boleh bergantung pada peluang, seperti kontrol industri, sinkronisasi perangkat keras, dan protokol keselamatan. Kelemahannya, biaya komunikasi dapat tinggi karena pihak-pihak yang terlibat harus menukar informasi cukup banyak untuk menghilangkan semua ketidakpastian. Analisis deterministik berguna sebagai batas dasar: jika suatu masalah mahal bahkan dalam model deterministik sederhana, maka sistem nyata membutuhkan strategi optimasi yang cermat.

3.3 KOMPLEKSITAS KOMUNIKASI DETERMINISTIK

Komunikasi deterministik menuntut bahwa protokol selalu menghasilkan keluaran benar untuk setiap pasangan input. Model ini cocok untuk sistem yang tidak boleh bergantung pada peluang, seperti kontrol industri, sinkronisasi perangkat keras, dan protokol keselamatan. Kelemahannya, biaya komunikasi dapat tinggi karena pihak-pihak yang terlibat harus menukar informasi cukup banyak untuk menghilangkan semua ketidakpastian. Analisis deterministik



berguna sebagai batas dasar: jika suatu masalah mahal bahkan dalam model deterministik sederhana, maka sistem nyata membutuhkan strategi optimasi yang cermat.

Bagian ini menyajikan model dasar dan paling terstruktur dari kompleksitas komunikasi: *model deterministik*. Dalam kerangka kerja ini, semua langkah komunikasi telah ditentukan sebelumnya, dan tidak ada keacakan atau toleransi kesalahan yang diizinkan. Tujuannya adalah untuk menganalisis jumlah minimum bit yang harus dipertukarkan oleh dua entitas komputasi untuk menghitung fungsi Boolean dengan kepastian penuh. Dari perspektif teknik, model ini sangat mencerminkan protokol komunikasi digital dunia nyata dalam sistem tertanam, jaringan sensor, dan arsitektur kontrol terdistribusi, di mana perilaku deterministik sangat penting untuk kebenaran, keamanan, dan jaminan waktu.

Pertimbangkan contoh kanonik: dua agen, Alice dan Bob, yang mewakili unit pemrosesan terdistribusi dalam suatu sistem (misalnya, mikrokontroler jaringan atau perangkat tepi), dan ingin mengidentifikasi slot waktu yang tersedia bersama—analog dengan tugas tingkat sistem seperti negosiasi sumber daya atau sinkronisasi. Masing-masing pihak memegang vektor biner n -bit lokal x dan y , masing-masing, menunjukkan slot waktu yang tersedia. Fungsi global yang akan dihitung adalah

$$f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{T, F\}$$

di mana $f(x, y) = T$ jika dan hanya jika ada indeks i sedemikian sehingga $x_i = y_i = 1$. Ini dikenal sebagai *Masalah Irisan Himpunan*, sebuah tolok ukur representatif dalam kompleksitas komunikasi.

Input x dan y diinterpretasikan sebagai vektor karakteristik dari himpunan bagian dari $[n] = \{1, \dots, n\}$, dan fungsi $f(x, y)$ menentukan apakah himpunan bagian ini beririsan. Dalam sistem terdistribusi, kueri semacam itu memodelkan operasi penggabungan dalam basis data terdistribusi, protokol kesepakatan, dan deteksi tabrakan di lapisan MAC. Untuk mendefinisikan secara formal protokol komunikasi deterministik, kita menggunakan representasi pohon keputusan biner:

Definisi 3.1

Protokol deterministik untuk $f: X \times Y \rightarrow \{T, F\}$ adalah pohon biner berakar T , dimana

- Setiap simpul internal diberi label dengan fungsi $a_v: X \rightarrow \{0,1\}$ atau $b_v: Y \rightarrow \{0,1\}$
- setiap daun diberi label keluaran dalam $\{T, F\}$.

Untuk setiap pasangan input (x, y) , jalur unik dari akar ke daun dilintasi berdasarkan hasil kueri di simpul internal: pada simpul v , protokol bergerak ke kiri jika $a_v(x) = 0$ atau $b_v(y) = 0$, jika tidak, ia bergerak ke kanan. Protokol menghitung f dengan benar jika untuk semua (x, y) label keluaran dari daun sama dengan $f(x, y)$. **Biaya** dari protokol adalah kedalaman pohon, yaitu jumlah maksimum bit yang dipertukarkan dalam jalur eksekusi apa pun. **Kompleksitas komunikasi deterministik** dari f , yang dilambangkan dengan $D(f)$, adalah biaya minimum dari semua protokol deterministik yang benar untuk f . Konsep kombinatorial sentral



dalam protokol deterministik adalah konsep *persegi panjang kombinatorial*. Konsep ini memfasilitasi hubungan antara kompleksitas komunikasi dan representasi berbasis matriks yang digunakan dalam pemrosesan sinyal dan teori jaringan.

Definisi 3.2

Persegi panjang kombinatorial adalah subset $C \subseteq X \times Y$ sedemikian sehingga jika $C = X_0 \times Y_0$ untuk suatu $X_0 \subseteq X$ dan $Y_0 \subseteq Y$. Setiap persegi panjang dapat dipandang sebagai matriks Boolean berperingkat satu R dengan ukuran $|X| \times |Y|$, di mana $R[x, y] = 1$ jika $(x, y) \in C$, dan $R[x, y] = 0$ jika sebaliknya. **Persegi panjang monokromatik** memainkan peran kunci dalam merancang dan menganalisis protokol deterministik:

- Sebuah persegi panjang C dikatakan monokromatik terhadap f jika $f(x, y)$ bernilai konstan untuk semua $(x, y) \in C$.
- Setiap protokol deterministik mempartisi ruang input $X \times Y$ ke dalam himpunan persegi panjang monokromatik seperti itu.

Definisi 3.3 (Bilangan Partisi)

Misalkan X dan Y adalah himpunan berhingga dan $f: X \times Y \rightarrow \{T, F\}$. **Bilangan partisi** $C_D(f)$ adalah jumlah minimum persegi panjang kombinatorial monokromatik yang diperlukan untuk mempartisi $X \times Y$ terhadap f .

Memahami sifat-sifat struktural ini tidak hanya memungkinkan kita untuk membatasi $D(f)$ dari bawah tetapi juga menjembatani kompleksitas komunikasi dengan peringkat matriks, kedalaman sirkuit, dan latensi komunikasi perangkat keras. Wawasan seperti itu sangat penting ketika merancang sistem desain bersama perangkat keras-perangkat lunak yang efisien, mengoptimalkan komunikasi bus, atau menerapkan protokol yang sadar bandwidth di platform tertanam jaringan.

3.4 KOMUNIKASI NONDETERMINISTIK

Komunikasi nondeterministik memperkenalkan gagasan sertifikat atau saksi yang membantu dua pihak memverifikasi jawaban. Model ini tidak selalu dimaksudkan sebagai mekanisme langsung di perangkat nyata, tetapi sangat berguna untuk memahami perbedaan antara menghitung dan memverifikasi. Dalam banyak sistem, bukti ringkas dapat membuat verifikasi jauh lebih murah daripada menemukan solusi. Ide ini muncul dalam validasi transaksi, protokol kriptografi, verifikasi identitas, dan sistem terdistribusi yang mengandalkan bukti kebenaran.

3.5 KOMUNIKASI NONDETERMINISTIK

Mirip dengan kelas kompleksitas komputasi klasik, kompleksitas komunikasi memiliki varian *nondeterministik* yang memodelkan komputasi di mana verifikasi lebih mudah daripada komputasi itu sendiri. Kompleksitas komunikasi nondeterministik sangat elegan dari perspektif kombinatorial dan, dalam beberapa hal, lebih baik dikarakterisasi dan dianalisis daripada rekan



deterministiknya. Ini juga mencontohkan "paradigma tiga langkah" komunikasi: menebak, verifikasi, dan kesimpulan. Secara formal, misalkan $f: X \times Y \rightarrow \{T, F\}$ adalah fungsi Boolean, dan definisikan himpunan contoh yang *benar* sebagai

$$L = \{(x, y) \in X \times Y: f(x, y) = T\}$$

Protokol komunikasi nondeterministik untuk f didefinisikan melalui dua fungsi verifikasi:

$$A: X \times \{0,1\}^k \rightarrow \{0,1\}, \quad B: Y \times \{0,1\}^k \rightarrow \{0,1\}$$

di mana $z \in \{0,1\}^k$ adalah sertifikat (atau "saksi") yang dipilih secara nondeterministik yang dibagikan kepada kedua pihak. Protokol tersebut memenuhi kondisi kebenaran berikut:

1. Untuk setiap $(x, y) \in L$, terdapat sertifikat z sedemikian sehingga $A(x, z) = 1$ dan $B(y, z) = 1$.
2. Untuk setiap $(x, y) \notin L$, untuk semua z , baik $A(x, z) = 0$ atau $B(y, z) = 0$.

Biaya protokol didefinisikan sebagai k , yaitu panjang sertifikat z . **Kompleksitas komunikasi nondeterministik** dari f dalam arah yang benar dilambangkan dengan $N_1(f)$ —nilai minimal dari k dimana protokol tersebut ada. Kompleksitas yang sesuai untuk contoh yang salah didefinisikan sebagai $N_0(f) = N_1(-f)$, dan kompleksitas nondeterministik keseluruhan adalah $N(f) = \max\{N_0(f), N_1(f)\}$. Formulasi alternatif dan setara memandang protokol nondeterministik dalam dua tahap:

1. *Pembukti* menyarankan sertifikat z kepada Alice dan Bob.
2. Alice dan Bob menggunakan input lokal mereka dan z untuk secara independen memverifikasi kebenaran menggunakan protokol deterministik atau logika verifikasi lokal.

Dari sudut pandang sistem, ini analog dengan verifikasi terdistribusi dalam sistem toleransi kesalahan atau verifikasi kriptografi berbasis saksi dalam bukti tanpa pengetahuan, di mana sertifikat bertindak sebagai saksi bersama atas kebenaran.

Contoh—Masalah Irisan Himpunan: Mari kita tinjau kembali fungsi *Irisan Himpunan* $f: \{0,1\}^n \times \{0,1\}^n \rightarrow \{T, F\}$, di mana vektor input x dan y mewakili vektor karakteristik dari himpunan bagian dari $[n]$. Misalkan pembukti pihak ketiga menyarankan indeks $i \in [n]$ di mana $x_i = y_i = 1$, mengklaim bahwa slot ini saling bebas. Alice dan Bob dapat secara independen memeriksa klaim ini hanya menggunakan $\log n$ bit untuk mengkodekan indeks. Oleh karena itu, kompleksitas komunikasi nondeterministik memenuhi:

$$N_1(f) \leq \log n$$

Hal ini menggambarkan bagaimana keberadaan sertifikat ringkas (indeks umum) secara dramatis mengurangi biaya komunikasi, dibandingkan dengan protokol deterministik, yang mungkin memerlukan $\Theta(n)$ bit dalam kasus terburuk. Model nondeterministik dengan



demikian menawarkan abstraksi yang ampuh untuk menganalisis tugas komunikasi berbasis verifikasi dalam sistem terdistribusi dan jaringan.

3.6 KOMPLEKSITAS KOMUNIKASI ACAK

Kompleksitas komunikasi acak menunjukkan bahwa toleransi terhadap kesalahan kecil dapat menurunkan kebutuhan komunikasi. Dalam jaringan nyata, pendekatan ini relevan ketika biaya mengirim semua informasi terlalu tinggi, sementara estimasi probabilistik sudah cukup untuk keputusan operasional. Contohnya adalah deteksi anomali terdistribusi, agregasi data sensor, dan estimasi frekuensi pada data streaming. Namun, penggunaan keacakan harus disertai batas kesalahan eksplisit, karena keputusan yang salah dapat berdampak berbeda bergantung pada konteks aplikasi.

Dalam protokol komunikasi acak, Alice dan Bob diizinkan untuk memanfaatkan keacakan selama interaksi mereka. Ini berarti bahwa pesan yang mereka pertukarkan mungkin bergantung tidak hanya pada input masing-masing dan komunikasi sebelumnya tetapi juga pada hasil peristiwa acak, seperti lemparan koin. Akibatnya, perilaku dan output protokol dapat bervariasi di berbagai eksekusi pada pasangan input yang sama. Untuk memperhitungkan variabilitas ini, persyaratan ketat untuk selalu menghasilkan output yang benar dilonggarkan. Sebagai gantinya, kita mengizinkan protokol untuk melakukan kesalahan dengan probabilitas kecil ϵ , dan mensyaratkan bahwa fungsi f dihitung dengan benar dengan probabilitas setidaknya $1 - \epsilon$ untuk semua input. Pelonggaran ini seringkali menghasilkan protokol dengan biaya komunikasi yang jauh lebih rendah untuk masalah tertentu. Namun, pengenalan keacakan juga mempersulit proses penetapan batas bawah yang ketat pada kompleksitas komunikasi. Kompleksitas komunikasi acak hadir dalam beberapa varian, yang dikategorikan ke dalam dua dimensi utama:

- Model Kesalahan:
 - **Protokol tanpa kesalahan (Las Vegas)** selalu menghasilkan hasil yang benar, meskipun biaya komunikasi dapat bervariasi.
 - **Protokol kesalahan satu sisi** hanya memungkinkan keluaran yang salah dalam satu arah (misalnya, positif palsu atau negatif palsu).
 - Protokol kesalahan dua sisi memungkinkan kesalahan pada kedua arah, tetapi probabilitas kesalahan total harus dibatasi oleh ϵ .
- Akses ke Keacakan:
 - **Protokol koin pribadi** menggunakan bit acak independen yang hanya dapat diakses oleh masing-masing pihak.
 - **Protokol koin publik** mengasumsikan sumber keacakan bersama yang diketahui oleh Alice dan Bob.

Dalam bab ini, kita fokus pada **model koin publik dengan kesalahan dua sisi**. Dalam model ini, Alice dan Bob memulai dengan akses ke string acak umum $r \in \{0,1\}^*$. Panjang string ini tidak terbatas dan tidak berkontribusi pada biaya komunikasi. Berdasarkan nilai r , mereka menjalankan protokol deterministik P_r yang perilakunya dapat bervariasi tergantung pada realisasi string acak.



Secara formal, protokol acak P dalam model koin publik dikatakan menghitung fungsi Boolean $f: X \times Y \rightarrow \{T, F\}$ dengan kesalahan paling banyak ϵ jika:

$$\Pr_r[P_r(x, y) = f(x, y)] \geq 1 - \epsilon \quad \text{for all } (x, y) \in X \times Y$$

Biaya komunikasi dari protokol semacam itu didefinisikan sebagai jumlah bit maksimum yang dipertukarkan dalam protokol deterministik P_r apa pun, di atas semua string acak r yang mungkin. **Kompleksitas komunikasi acak** dari f dengan kesalahan ϵ dilambangkan dengan $R_\epsilon(f)$ dan mewakili biaya minimum tersebut di atas semua protokol acak koin publik yang mencapai jaminan kebenaran yang diperlukan.

Peringkat Perkiraan

Peringkat perkiraan menghubungkan kompleksitas komunikasi dengan aljabar linear. Ide dasarnya adalah bahwa struktur matriks fungsi dapat mencerminkan jumlah informasi yang perlu dikomunikasikan. Jika matriks dapat didekati oleh matriks berpangkat rendah, maka ada indikasi bahwa fungsi tersebut memiliki struktur yang dapat dieksploitasi untuk komunikasi yang lebih hemat. Teknik ini penting dalam pembuktian batas bawah, tetapi secara komputasional dapat sulit karena masalah minimisasi peringkat sering memiliki ruang pencarian besar dan kendala yang kompleks. Salah satu teknik batas bawah dasar dalam kompleksitas komunikasi—yang berlaku untuk pengaturan deterministik dan acak—didasarkan pada peringkat matriks. Dalam model acak, konsep ini diperluas ke apa yang dikenal sebagai *peringkat perkiraan*, yang menangkap peringkat minimal dari matriks yang mendekati matriks tanda yang diberikan.

Definisi 3.4 (Peringkat Aproksimasi).

Misalkan A adalah matriks tanda, yaitu matriks dengan entri dalam $\{-1, +1\}$. Peringkat perkiraan dari A dengan faktor pendekatan $\alpha > 1$, yang dilambangkan dengan $\text{rank}_\alpha(A)$, didefinisikan sebagai:

$$\text{rank}_\alpha(A) = \min\{\text{rank}(B) : 1 \leq A[i, j] \cdot B[i, j] \leq \alpha \text{ for all } (i, j)\}$$

di mana B berkisar pada matriks bernilai riil dengan dimensi yang sama seperti A . Ketika faktor pendekatan α menuju tak hingga, diperoleh ukuran terkait yang dikenal sebagai **peringkat tanda**:

$$\text{rank}^\infty(A) = \min\{\text{rank}(B) : A[i, j] \cdot B[i, j] \geq 1 \text{ for all } (i, j)\}$$

Kuantitas ini mencerminkan peringkat terkecil dari matriks riil B yang mempertahankan pola tanda A . Penting untuk dicatat bahwa menghitung peringkat perkiraan melibatkan penyelesaian masalah optimasi, yang seringkali menantang dalam praktiknya. Meskipun masih belum jelas apakah menghitung $\text{rank}_\alpha(A)$ adalah NP-hard untuk α secara umum, kelas masalah minimisasi peringkat yang lebih luas di bawah kendala linier diketahui mencakup banyak contoh NP-hard.



3.7 LATIHAN

1. Kembangkan protokol komunikasi deterministik untuk *Masalah Keterpisahan Himpunan*. Mulailah dengan merancang protokol dan mengimplementasikannya menggunakan bahasa pemrograman pilihan Anda. Analisis jumlah bit yang dipertukarkan antara dua pihak, Alice dan Bob, untuk berbagai ukuran input, dan evaluasi biaya komunikasinya. Buat laporan yang mencakup protokol Anda, hasil implementasi, dan diskusi tentang kekuatan dan keterbatasan pendekatan Anda.
2. Pilih masalah komunikasi nondeterministik seperti *Masalah Kesetaraan* atau *Masalah Irisan Himpunan*. Kembangkan protokol komunikasi nondeterministik dan jelaskan cara kerjanya secara detail. Implementasikan protokol dan evaluasi efisiensinya dalam hal kompleksitas komunikasi. Bandingkan hasilnya dengan protokol deterministik untuk masalah yang sama dan soroti perbedaannya dalam hal biaya komunikasi dan beban komputasi.
3. Rancang dan implementasikan protokol komunikasi acak untuk tugas komputasi terdistribusi, seperti *Pemilihan Pemimpin* atau *Pemungutan Suara Mayoritas*. Manfaatkan model koin publik untuk mengurangi jumlah bit yang dipertukarkan antar node. Analisis probabilitas kesalahan protokol Anda, serta kompleksitas komunikasi total. Siapkan laporan yang mencakup analisis teoretis protokol Anda, hasil empiris, dan saran untuk potensi perbaikan.
4. Jelajahi bagaimana prinsip kompleksitas komunikasi dapat diterapkan untuk mengoptimalkan perutean jaringan. Kembangkan protokol perutean yang meminimalkan beban komunikasi dalam lingkungan jaringan simulasi. Lakukan eksperimen untuk membandingkan kinerja protokol Anda dengan algoritma perutean standar. Sajikan temuan Anda dalam makalah bergaya pengembangan ilmiah, menekankan pertukaran komunikasi dan keunggulan pendekatan Anda dibandingkan metode tradisional.
5. Pilih masalah yang terkait dengan peringkat perkiraan, seperti *Perkalian Matriks* atau *Pencocokan Graf*, dan selidiki bagaimana peringkat perkiraan memengaruhi kompleksitas komunikasi untuk masalah ini. Implementasikan algoritma untuk memperkirakan peringkat perkiraan matriks untuk berbagai faktor perkiraan. Tulis laporan yang membahas metodologi Anda, hasil eksperimen, dan implikasi peringkat perkiraan terhadap kompleksitas komunikasi. Selain itu, berikan saran tentang bagaimana memanfaatkan peringkat perkiraan untuk protokol komunikasi yang lebih efisien.



BAB 4

KOMPLEKSITAS DATA

4.1 TEORI INFORMASI ALGORITMA

Bab ini memperluas konsep kompleksitas ke ranah data. Dalam sistem cerdas, kualitas keputusan algoritmik sangat bergantung pada struktur, keteraturan, keragaman, dan tingkat kebisingan data. Data yang besar belum tentu informatif; sebaliknya, data yang lebih kecil tetapi terstruktur dapat menghasilkan model yang lebih stabil. Kompleksitas data membantu menjelaskan mengapa model tertentu mudah mengalami overfitting, mengapa dataset acak sulit dikompresi, dan mengapa ukuran seperti kompleksitas Kolmogorov, dimensi VC, serta kompleksitas Rademacher penting untuk memahami hubungan antara kapasitas model dan kemampuan generalisasi.

Teori informasi algoritmik mempelajari kompleksitas string data. String yang kompleks sulit dikompresi. Meskipun kompresi bergantung pada codec, bahasa Turing-lengkap dapat menerjemahkan satu sama lain. Untuk string yang besar, overhead penerjemahan menjadi dapat diabaikan. Kompleksitas algoritmik sering melabeli noise acak sebagai kompleks, yang kontras dengan pemahaman kompleksitas dalam sistem kompleks. Entropi informasi juga mengukur kompleksitas, tetapi juga memberikan kompleksitas tinggi pada keacakan. Namun, kompleksitas fluktuasi informasi menghindari masalah ini dengan berfokus pada fluktuasi entropi, yang terbukti berguna dalam aplikasi. Penelitian pembelajaran mesin (ML) mengeksplorasi bagaimana kompleksitas data memengaruhi klasifikasi terawasi. Ho dan Basu mengusulkan ukuran kompleksitas untuk klasifikasi biner (Gambar 4.1), yang berfokus pada:



Gambar 4.1 Berbagai Penyebab Kompleksitas Data.

- Tumpang tindih nilai fitur antar kelas.
- Keterpisahan kelas-kelas.
- Sifat geometris, topologis, dan densitas dari manifold.
- Kesulitan instans, yang digeneralisasikan ke masalah non-biner, menilai tingkat kesulitan klasifikasi.

Kekerasan instans, yang digeneralisasikan ke masalah non-biner, menilai kesulitan klasifikasi.



Teori informasi algoritma berupaya mengukur informasi berdasarkan panjang program terpendek yang menghasilkan suatu objek. Dengan pendekatan ini, keteraturan berarti dapat dikompresi, sedangkan keacakan berarti tidak memiliki deskripsi yang jauh lebih pendek daripada objek itu sendiri. Konsep ini memperkaya teori informasi klasik karena tidak hanya bergantung pada distribusi probabilitas, tetapi pada kemampuan deskriptif algoritmik. Dalam data modern, gagasan ini membantu memahami mengapa beberapa pola mudah dipelajari dan beberapa data tampak acak bagi model. Teori informasi algoritma berfokus pada beberapa aspek kunci, termasuk kompleksitas Kolmogorov, informasi timbal balik algoritma, hubungannya dengan entropi dan informasi timbal balik Shannon, statistik cukup minimal algoritma, dan fungsi struktur Kolmogorov. Konsep-konsep ini selanjutnya dikaitkan dengan gagasan informasi yang bermakna. Sepanjang bab ini, kami menekankan wawasan yang, menurut pandangan kami, bersifat filosofis. Sekarang saatnya untuk mengkonsolidasikan ide-ide ini dan membuat hubungan filosofisnya eksplisit.

Di bawah ini, pertama-tama kita membahas beberapa implikasi teori informasi algoritma untuk filsafat matematika, teori probabilitas, dan statistik. Kemudian kita menyimpulkan bab ini dengan memeriksa implikasi filosofis untuk konsep informasi itu sendiri. Seperti yang akan kita lihat, hampir semua implikasi filosofis ini terkait dengan gagasan tentang keacakan.

Filsafat Matematika: Keacakan Dalam Matematika

Dalam filsafat matematika, keacakan menjadi persoalan yang menarik karena objek matematika dapat memiliki struktur tersembunyi yang tidak segera terlihat. Suatu urutan angka dapat tampak acak secara visual, tetapi sebenarnya dihasilkan oleh aturan singkat. Sebaliknya, urutan pendek dapat tampak berpola tetapi tidak memiliki deskripsi umum yang sederhana. Teori informasi algoritma memberi alat untuk membedakan keteraturan sejati dan keteraturan semu, meskipun pengukuran pastinya sering tidak dapat dihitung secara umum. Prinsip-prinsip yang mendasari kompleksitas Kolmogorov memiliki hubungan yang mendalam dengan teorema ketidaklengkapan Gödel. Secara khusus, setiap sistem aksioma formal dengan kompleksitas Kolmogorov terbatas pasti mengarah pada persamaan $-K(X) = \min_{p \in U(p)} \{ |p| \}$, di mana $|p|$ adalah panjang program p dan $U(p)$ adalah mesin universal yang menerima program p —yang solusinya menunjukkan karakteristik keacakan algoritmik. Persamaan-persamaan ini tidak dapat diselesaikan secara menyeluruh oleh sistem aksioma yang dapat dihitung, yang menyiratkan suatu bentuk ketidaklengkapan yang didorong oleh keacakan. Dalam konteks ini, Chaitin terkenal menyatakan:

Ini adalah wilayah di mana kebenaran matematika tidak memiliki struktur atau pola yang dapat dikenali dan tampak sepenuhnya acak.

Meskipun mekanika kuantum telah menunjukkan bahwa keacakan ada di dunia fisik, keacakan serupa juga muncul dalam ranah matematika murni. Pengamatan ini tidak menyiratkan kerusakan tatanan matematika; melainkan, menunjukkan hukum probabilitas atau statistik yang mengatur domain tertentu. Hal ini mengundang perspektif di mana teori



bilangan—dan mungkin matematika secara umum—dapat memperoleh manfaat dari pendekatan yang lebih empiris atau eksploratif, mirip dengan metodologi ilmu alam.

Filsafat Probabilitas: Memahami Keacakan Urutan Individual

Filsafat probabilitas menanyakan bagaimana kita memahami keacakan pada satu urutan individual, bukan hanya pada kumpulan kejadian berulang. Pendekatan statistik tradisional biasanya berbicara tentang distribusi, sedangkan kompleksitas Kolmogorov mencoba menilai apakah sebuah objek individual dapat dikompresi. Perbedaan ini penting dalam analisis data karena dataset tunggal sering menjadi dasar keputusan. Jika dataset tampak acak tetapi sebenarnya memiliki struktur, model dapat belajar; jika benar-benar tidak terkompresi, kemampuan prediksi akan sangat terbatas.

Dalam teori probabilitas klasik, pernyataan seperti “ x adalah urutan acak” pada dasarnya tidak terdefinisi dengan baik. Kerangka kerja tradisional membahas koleksi atau himpunan, memberikan hasil yang berlaku hampir pasti atau dengan probabilitas tinggi atas banyak hasil. Namun, dalam praktiknya, kita hanya mengamati satu urutan. Apa artinya menyatakan bahwa “urutan ini tipikal di bawah distribusi P ”? Seseorang mungkin tergoda untuk menafsirkan ini sebagai urutan yang memenuhi semua sifat yang terjadi dengan probabilitas satu. Namun, gagasan ini gagal karena irisan dari semua himpunan tersebut kosong. Teori keacakan algoritmik—yang dipelopori oleh Martin-Löf—menawarkan solusi. Dalam kerangka kerja ini, suatu urutan dianggap acak bagi P jika tidak dapat dibedakan secara efektif dari keacakan oleh uji komputasi apa pun. Definisi ini terkait erat dengan kompleksitas Kolmogorov: suatu urutan x dikatakan acak jika kompleksitas Kolmogorovnya $K(x)$ kira-kira sama dengan panjangnya $l(x)$; dengan kata lain, urutan tersebut tidak dapat dikompresi oleh algoritma apa pun.

Pendekatan ini memungkinkan kita untuk membahas secara ketat keacakan urutan individual—sesuatu yang tidak mungkin dilakukan dalam probabilitas klasik. Yang penting, definisi Martin-Löf selaras dengan penalaran probabilistik konvensional, menjembatani kesenjangan antara pandangan algoritmik dan klasik. Berbagai bentuk keacakan (misalnya, keacakan positif vs. negatif) dapat diformalkan dalam kerangka kerja ini—konsep di luar jangkauan bahasa probabilistik tradisional.

4.2 PRINSIP OCCAM DAN INFERENSI INDUKTIF

Prinsip Occam menyatakan bahwa penjelasan yang lebih sederhana sebaiknya dipilih ketika memiliki daya jelaskan yang memadai. Dalam pembelajaran mesin, prinsip ini berkaitan dengan pemilihan model: model yang terlalu kompleks dapat menghafal data latih, sedangkan model yang terlalu sederhana dapat gagal menangkap pola penting. Kompleksitas algoritmik memberi dasar formal bagi prinsip ini melalui gagasan deskripsi pendek. Namun, kesederhanaan harus dinilai bersama akurasi, konteks, dan risiko kesalahan, bukan dijadikan aturan mutlak. Teori informasi algoritmik berkontribusi pada penalaran induktif melalui dua perspektif yang saling melengkapi: teori statistik yang cukup algoritmik, prinsip panjang deskripsi minimum (MDL), dan teori induksi universal Solomonoff. Meskipun yang terakhir tidak dijelaskan secara rinci di sini, teori ini memainkan peran mendasar dalam mempelajari



pemodelan prediktif. Baik MDL maupun induksi Solomonoff mencerminkan interpretasi komputasional dari Prinsip Occam: ketika dihadapkan dengan beberapa hipotesis yang menjelaskan data dengan sama baiknya, preferensi diberikan kepada hipotesis dengan deskripsi paling sederhana. Prinsip MDL menekankan penemuan struktural dalam data, sementara pendekatan Solomonoff berfokus pada prediksi urutan.

Banyak teorema konvergensi membenarkan penggunaan kerangka kerja ini. Misalnya, jika proses pembangkitan data aktual termasuk dalam kelas model yang diketahui, maka MDL dijamin dapat mengidentifikasi model ini dari data yang terbatas. Hal ini berlaku untuk kedua varian praktis—di mana M adalah kelas model yang dibatasi dan varian teoritis berdasarkan kompleksitas Kolmogorov, di mana M adalah kelas distribusi probabilitas yang dapat dihitung. Keberhasilan metode ini bergantung pada biasnya terhadap kesederhanaan. Secara khusus, konvergensi bergantung pada penggunaan ukuran kompleksitas yang valid—biasanya panjang kode $L(P)$ setiap model di bawah kode bebas awalan. Jika kondisi ini tidak terpenuhi, konvergensi dapat gagal, dan model yang terlalu kompleks dapat dipilih, bahkan dengan data yang meningkat. Menariknya, bahkan ketika model sebenarnya sangat kompleks, mengutamakan kesederhanaan justru menguntungkan ketika ukuran sampel kecil. Wawasan ini secara formal membenarkan penggunaan varian Pisau Cukur Occam MDL dalam inferensi induktif. Namun, penting untuk memahami bahwa ini adalah prinsip epistemologis—sebuah heuristik praktis—bukan klaim tentang kesederhanaan inheren dunia.

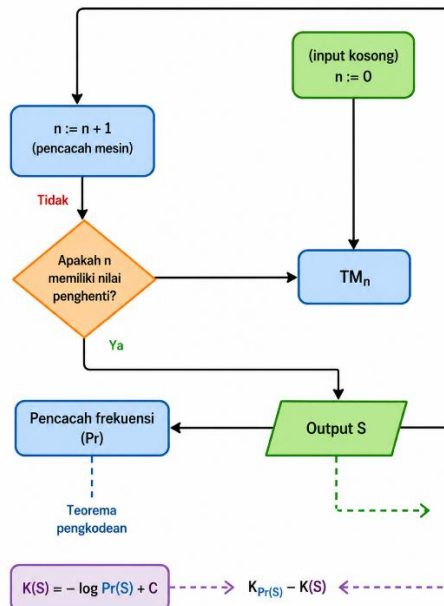
4.3 FILSAFAT INFORMASI

Filsafat informasi membahas apa yang membuat suatu data menjadi informasi yang bermakna. Tidak semua urutan simbol memiliki relevansi terhadap dunia. Data dapat memiliki kompleksitas tinggi tetapi tidak berguna jika tidak terkait dengan pertanyaan, konteks, atau tujuan tertentu. Dalam sistem AI, isu ini tampak pada perbedaan antara korelasi statistik dan pengetahuan operasional. Model dapat menemukan pola numerik, tetapi manusia tetap perlu menilai apakah pola tersebut bermakna, dapat dijelaskan, dan layak digunakan dalam keputusan. Dalam konsep pengantar teori informasi, “informasi selalu merupakan informasi tentang sesuatu.” Hal ini selaras dengan kerangka kerja Shannon, di mana sebuah pesan mengkodekan informasi tentang sumber eksternal. Namun, ketika mengukur kandungan informasi dari sebuah string x menggunakan kompleksitas Kolmogorov $K(x)$, sifat informasi menjadi kurang referensial secara eksternal. Informasi yang berkaitan dengan struktur intrinsik x itu sendiri tidak selalu terkait dengan realitas eksternal apa pun.

Sebagai contoh, pertimbangkan satu miliar digit pertama dari ekspansi biner π . Meskipun urutan ini dapat dihasilkan oleh program pendek dan karenanya memiliki kompleksitas Kolmogorov yang rendah, hal itu mungkin tidak menyampaikan pengetahuan baru tentang dunia eksternal. Hal ini menimbulkan pertanyaan filosofis tentang apakah informasi tersebut dianggap sebagai “tentang” sesuatu. Ambiguitas ini telah menyebabkan perdebatan di dalam komunitas. Misalnya, peserta mempertanyakan apakah kompleksitas algoritmik benar-benar dapat mewakili informasi selama lokakarya, mengingat kurangnya konten referensial. Meskipun demikian, contoh praktis menunjukkan sebaliknya. Perhatikan



kompleksitas bersyaratnya: $K(y | x)$: jika y adalah tiga kali x , keduanya dinyatakan dalam biner, dan x adalah string n -bit acak (sehingga $K(x) \approx K(y) \approx n$), maka $K(y | x) = O(1)$. Ini menyiratkan bahwa sebagian besar informasi yang diperlukan untuk menggambarkan y sudah dikodekan dalam x , menunjukkan bentuk ketergantungan informasional bahkan dalam pengaturan algoritmik (Gambar 4.2).



Gambar 4.2 Algoritma Pengkodean Informasi Untuk Kompleksitas Data.

4.4 PELAJARAN UNTUK FILSAFAT INFORMASI

Pelajaran utama dari teori informasi algoritma adalah bahwa informasi dapat diukur dari sisi struktur internal data, tetapi makna memerlukan relasi dengan konteks. Hal ini penting untuk ilmu data modern. Kompresibilitas dapat menunjukkan adanya pola, tetapi tidak otomatis menunjukkan relevansi. Karena itu, analisis informasi perlu menggabungkan ukuran formal dengan interpretasi domain. Pendekatan ini membantu menghindari kesalahan umum: menganggap data besar atau kompleks selalu lebih bernilai daripada data sederhana yang tepat sasaran. Dari teori informasi algoritmik, kita dapat menarik beberapa pelajaran untuk filsafat informasi:

1. **Pengukuran Informasi Objektif:** Jumlah absolut informasi dalam suatu urutan dapat didefinisikan dengan baik secara asimtotik tetapi tetap umumnya tidak dapat dihitung. Oleh karena itu, pengukuran informasi objektif tanpa referensi eksternal dapat dicapai dengan mengorbankan kemampuan komputasi umum.
2. **Informasi Bermakna:** Teori ini memungkinkan formalisasi baru dari informasi bermakna. Teori ini dapat memisahkan bagian data yang bermakna dari noise tanpa bergantung pada asumsi probabilistik. Pemisahan ini sangat penting untuk menghubungkan teori ini dengan inferensi induktif.
3. **Hubungan dengan Kompresi Data:** Teori informasi algoritmik menekankan kembali hubungan antara pengukuran informasi dan kompresi data, sebuah konsep dasar



dalam teori Shannon. Jika data x dihasilkan oleh beberapa proses probabilistik P , maka informasi algoritmik dalam x berperilaku serupa dengan entropi Shannon dari P .

4.5 METRIK KEACAKAN KOLMOGOROV

Kompleksitas Kolmogorov memberikan definisi formal tentang keacakan melalui ketidakmampuan kompresi. Jika sebuah string dapat dihasilkan oleh program pendek, string tersebut memiliki struktur; jika tidak ada deskripsi yang lebih pendek daripada string itu sendiri, string tersebut dianggap kompleks atau acak. Dalam praktik, kompleksitas Kolmogorov tidak dapat dihitung secara tepat untuk semua objek, tetapi dapat diperkirakan melalui kompresi. Konsep ini berguna untuk memahami data, model, dan batas kemampuan prediksi. Kompleksitas Kolmogorov membahas bagaimana mendefinisikan keacakan untuk urutan terbatas. Laplace mencatat bahwa urutan sederhana memiliki aturan yang mudah dipahami, sementara urutan kompleks dan tidak beraturan dianggap acak. Teori ini mengukur kandungan informasi dari sebuah string berdasarkan panjang deskripsi terpendeknya. Deskripsi pendek menyiratkan kesederhanaan, sementara deskripsi panjang menunjukkan kompleksitas. Probabilitas algoritmik Solomonoff lebih lanjut menetapkan probabilitas berdasarkan kompleksitas string, membedakan string reguler dari string acak.

Mendefinisikan Deskripsi Dan Kompleksitas

Definisi deskripsi harus dibuat formal karena bahasa sehari-hari dapat menimbulkan paradoks. Dalam teori algoritmik, deskripsi biasanya dipahami sebagai program pada mesin universal tertentu. Program tersebut menerima input tertentu dan menghasilkan output yang ingin dijelaskan. Dengan cara ini, panjang deskripsi dapat diukur secara objektif. Meskipun pilihan mesin universal dapat memengaruhi konstanta, teorema invarian memastikan bahwa perbedaan tersebut tidak mengubah perilaku asimtotik ukuran kompleksitas. Untuk menghindari paradoks seperti paradoks Berry, kita membutuhkan definisi deskripsi yang tepat.

Paradoks Berry

Paradoks Berry menunjukkan bahaya menggunakan bahasa natural tanpa batas formal. Kalimat yang tampak sederhana dapat menciptakan kontradiksi karena merujuk pada batas deskripsi dirinya sendiri. Dalam teori kompleksitas, paradoks ini dihindari dengan membatasi deskripsi pada program formal dan mesin komputasi yang terdefinisi. Pelajaran metodologisnya adalah bahwa konsep seperti “pendek”, “sederhana”, dan “dapat dijelaskan” harus diberi definisi operasional sebelum digunakan sebagai dasar pengukuran ilmiah. Pertimbangkan: “Bilangan Berry adalah bilangan bulat positif terkecil yang tidak dapat dijelaskan dalam kurang dari delapan belas kata.” Definisi ini, menggunakan tujuh belas kata, menciptakan paradoks dengan mendeskripsikan bilangan yang diklaim tidak dapat dijelaskan.

Deskripsi Algoritma

Deskripsi algoritma mengubah gagasan penjelasan menjadi objek yang dapat dihitung: sebuah program. Jika program pendek dapat menghasilkan string panjang, maka string tersebut memiliki keteraturan. Misalnya, satu juta digit yang mengikuti pola berulang dapat dijelaskan oleh program jauh lebih pendek daripada menuliskan seluruh digitnya. Sebaliknya, string acak membutuhkan deskripsi hampir sepanjang string itu sendiri. Perbedaan ini menjadi



dasar hubungan antara kompleksitas, kompresi, dan pembelajaran. Misalkan P adalah sebuah program dan σ sebuah string biner. Sebuah string δ merupakan deskripsi- P dari σ jika P dengan input δ menghasilkan output σ . **Definisi: Kompleksitas algoritmik- P .** Kompleksitas- P dari σ , $CP(\sigma)$, adalah panjang deskripsi- P terpendek dari σ , atau ∞ jika tidak ada. Program dengan $CP(\sigma)$ yang terbatas untuk semua σ adalah "kompleksitas-terbatas."

Kompresi Dan Invarian

Kompresi menunjukkan bahwa suatu data memiliki struktur yang dapat dieksploitasi. Teorema invarian menjamin bahwa ukuran kompleksitas tidak sepenuhnya bergantung pada bahasa pemrograman atau mesin universal yang dipilih, karena perbedaannya hanya berupa konstanta. Ini penting agar kompleksitas Kolmogorov dapat dipakai sebagai ukuran teoritis yang stabil. Dalam praktik, algoritma kompresi digunakan sebagai pendekatan empiris, meskipun tidak selalu menangkap seluruh struktur algoritmik yang mungkin. Jika $CP(\sigma)$ jauh lebih pendek daripada $|\sigma|$, maka σ terkompresi dengan baik. Faktor kompresinya adalah $|\sigma|/C_P(\sigma)$.

Teorema Invarian

Teorema invarian memberikan dasar keuniversalan kompleksitas Kolmogorov. Selama mesin universal yang digunakan cukup kuat untuk mensimulasikan mesin lain, panjang deskripsi terpendek hanya berbeda sebesar konstanta tetap. Artinya, untuk objek yang sangat besar, perbedaan konstanta menjadi relatif tidak penting. Teorema ini membuat konsep kompleksitas algoritmik lebih kokoh karena tidak bergantung sepenuhnya pada pilihan representasi teknis tertentu. **Teorema: Teorema Invarian Solomonoff–Kolmogorov–Chaitin.** Terdapat program optimal U yang meminimalkan $C_U(\sigma)$ untuk semua σ . Semua program optimal setara hingga perbedaan konstan. Ini memastikan ukuran universal dari kompleksitas algoritma.

Keacakan Dan Kompresibilitas

Keacakan dalam teori ini dipahami sebagai ketidakmampuan untuk menemukan deskripsi yang lebih pendek. Data acak tidak berarti tidak memiliki nilai, tetapi berarti tidak memiliki pola yang dapat dimanfaatkan untuk kompresi atau prediksi. Dalam pembelajaran mesin, hal ini membantu menjelaskan mengapa noise murni tidak dapat dipelajari secara bermakna. Model yang tampak berhasil pada noise biasanya hanya menghafal, bukan menemukan struktur yang dapat digeneralisasikan. **Definisi: Keacakan dan Kompresibilitas.**

- σ dikatakan terkompresi- b jika $C(\sigma) \leq |\sigma| - b$.
- σ dikatakan terkompresi jika $C(\sigma) < |\sigma|$.
- σ dikatakan acak jika $C(\sigma) \geq |\sigma|$.

Ini menangkap intuisi Laplace tentang keacakan.

Hubungan Dengan Teorema Gödel

Keterkaitan dengan teorema Gödel menunjukkan bahwa ada batas formal terhadap apa yang dapat dibuktikan dalam sistem matematika. Dalam versi kompleksitas informasi, sebagian besar string memang acak, tetapi sistem formal tertentu hanya dapat membuktikan keacakan untuk sejumlah terbatas kasus. Ini memperlihatkan hubungan mendalam antara komputasi, bukti, dan informasi: tidak semua kebenaran atau sifat kompleksitas dapat



dijangkau oleh prosedur formal yang tetap. Meskipun sebagian besar string bersifat acak, hanya sejumlah string yang dapat dibuktikan acak. Ini berkaitan dengan teorema ketidaklengkapan Gödel. **Teorema: Teorema Ketidaklengkapan Gödel**, Versi Kompleksitas Informasi. Hanya sejumlah pernyataan $C(\sigma) = n$ yang dapat dibuktikan dalam suatu sistem formal.

4.6 METRIK KOMPLEKSITAS VC

Dimensi VC mengukur kapasitas kelas model untuk memisahkan data. Semakin tinggi dimensi VC, semakin fleksibel model dalam menyesuaikan berbagai pola pelabelan. Fleksibilitas ini bermanfaat jika pola data memang kompleks, tetapi berisiko menyebabkan overfitting jika data terbatas atau penuh noise. Dalam desain model pembelajaran mesin, dimensi VC membantu menjelaskan trade-off antara kapasitas, jumlah data yang dibutuhkan, dan kemampuan generalisasi. Dimensi Vapnik–Chervonenkis (VC) mengukur kompleksitas model klasifikasi. Dimensi ini mengukur berapa banyak titik data yang dapat diklasifikasikan dengan sempurna oleh model untuk setiap kemungkinan pelabelan. Konsep ini, yang diperkenalkan oleh Vapnik dan Chervonenkis, membantu memahami kemampuan model untuk menyesuaikan data. Secara intuitif, kompleksitas model berkaitan dengan fleksibilitasnya. Polinomial berderajat tinggi, misalnya, dapat menyesuaikan banyak titik data dengan sempurna tetapi mungkin berkinerja buruk pada data baru karena kompleksitasnya yang berlebihan. Model linier yang lebih sederhana mungkin tidak sesuai dengan data pelatihan dengan sempurna, tetapi dapat melakukan generalisasi dengan lebih baik.

Dimensi VC Dari Keluarga Himpunan

Dimensi VC mengukur kapasitas kelas model untuk memisahkan data. Semakin tinggi dimensi VC, semakin fleksibel model dalam menyesuaikan berbagai pola pelabelan. Fleksibilitas ini bermanfaat jika pola data memang kompleks, tetapi berisiko menyebabkan overfitting jika data terbatas atau penuh noise. Dalam desain model pembelajaran mesin, dimensi VC membantu menjelaskan trade-off antara kapasitas, jumlah data yang dibutuhkan, dan kemampuan generalisasi. Diberikan keluarga himpunan H dan himpunan C , irisan keduanya didefinisikan sebagai:

$$H \cap C = \{h \cap C \mid h \in H\}$$

Suatu himpunan C dikatakan "terpecah" oleh H jika $H \cap C$ mengandung semua himpunan bagian dari C , artinya $|H \cap C| = 2^{|C|}$. Dimensi VC dari H adalah ukuran himpunan C terbesar yang dapat dipecah oleh H . Jika himpunan yang sangat besar dapat terpecah, maka dimensi VC adalah tak terbatas.

Dimensi Vc Model Klasifikasi

Dimensi VC mengukur kapasitas kelas model untuk memisahkan data. Semakin tinggi dimensi VC, semakin fleksibel model dalam menyesuaikan berbagai pola pelabelan. Fleksibilitas ini bermanfaat jika pola data memang kompleks, tetapi berisiko menyebabkan overfitting jika data terbatas atau penuh noise. Dalam desain model pembelajaran mesin,



dimensi VC membantu menjelaskan trade-off antara kapasitas, jumlah data yang dibutuhkan, dan kemampuan generalisasi. Suatu model klasifikasi biner f memecah himpunan titik $(x_1, \dots, x_n)$ jika, untuk setiap kemungkinan pelabelan titik-titik ini, terdapat pengaturan parameter untuk f yang mengklasifikasikan semua titik dengan benar. Dimensi VC dari f adalah jumlah maksimum titik yang dapat terpecah. Lebih formalnya, ini adalah jumlah terbesar D sedemikian rupa sehingga terdapat himpunan D titik yang dapat dipecah oleh f .

4.7 KOMPLEKSITAS RADEMACHER

Kompleksitas Rademacher mengukur kemampuan kelas fungsi menyesuaikan label acak. Jika sebuah kelas model dapat menyesuaikan noise acak dengan sangat baik, maka kelas tersebut memiliki kapasitas tinggi dan berpotensi overfitting. Ukuran ini bersifat bergantung pada data, sehingga sering lebih informatif dibanding ukuran kapasitas yang hanya berbasis struktur model. Dalam teori pembelajaran statistik, kompleksitas Rademacher digunakan untuk menurunkan batas generalisasi yang menghubungkan performa data latih dengan performa populasi. Kompleksitas Rademacher adalah ukuran fundamental dalam teori pembelajaran statistik yang digunakan untuk menilai daya ekspresif kelas hipotesis. Ini mengukur seberapa baik kelas fungsi dapat menyesuaikan noise acak, dan memainkan peran sentral dalam menurunkan jaminan generalisasi untuk algoritma pembelajaran.

Kompleksitas Rademacher Dari Suatu Himpunan

Kompleksitas Rademacher mengukur kemampuan kelas fungsi menyesuaikan label acak. Jika sebuah kelas model dapat menyesuaikan noise acak dengan sangat baik, maka kelas tersebut memiliki kapasitas tinggi dan berpotensi overfitting. Ukuran ini bersifat bergantung pada data, sehingga sering lebih informatif dibanding ukuran kapasitas yang hanya berbasis struktur model. Dalam teori pembelajaran statistik, kompleksitas Rademacher digunakan untuk menurunkan batas generalisasi yang menghubungkan performa data latih dengan performa populasi. Misalkan $A \subseteq \mathbb{R}^m$ adalah himpunan vektor riil yang terbatas. Kompleksitas Rademacher dari A didefinisikan sebagai:

$$\text{Rad}(A) = \frac{1}{m} \mathbb{E} \left[\sup_{a \in A} \sum_{i=1}^m \sigma_i a_i \right]$$

di mana $\sigma_1, \dots, \sigma_m$ adalah variabel acak Rademacher independen dengan:

$$\Pr(\sigma_i = +1) = \Pr(\sigma_i = -1) = \frac{1}{2}$$

Beberapa versi definisi ini menyertakan nilai absolut, tetapi untuk himpunan simetris, ini tidak mengubah nilai ekspresi tersebut.

Kompleksitas Rademacher Dari Kelas Fungsi

Kompleksitas Rademacher mengukur kemampuan kelas fungsi menyesuaikan label acak. Jika sebuah kelas model dapat menyesuaikan noise acak dengan sangat baik, maka kelas



tersebut memiliki kapasitas tinggi dan berpotensi overfitting. Ukuran ini bersifat bergantung pada data, sehingga sering lebih informatif dibanding ukuran kapasitas yang hanya berbasis struktur model. Dalam teori pembelajaran statistik, kompleksitas Rademacher digunakan untuk menurunkan batas generalisasi yang menghubungkan performa data latih dengan performa populasi. Misalkan $S = (z_1, \dots, z_m) \in Z^m$ adalah sampel yang diambil dari domain Z , dan misalkan F adalah kelas fungsi bernilai riil yang didefinisikan pada Z . Kompleksitas Rademacher empiris dari F terhadap sampel S adalah

$$\text{Rad}_S(F) = \frac{1}{m} \mathbb{E} \left[\sup_{f \in F} \sum_{i=1}^m \sigma_i f(z_i) \right]$$

Hal ini dapat diungkapkan secara setara sebagai

$$\text{Rad}_S(F) = \text{Rad}(F \circ S)$$

Di mana

$$F \circ S = \{(f(z_1), \dots, f(z_m)) \mid f \in F\}$$

Misalkan P adalah distribusi probabilitas pada Z . Maka kompleksitas Rademacher yang diharapkan dari F untuk sampel berukuran m yang diambil secara i.i.d. dari P didefinisikan sebagai:

$$\text{Rad}_{P,m}(F) = \mathbb{E}_{S \sim P^m} [\text{Rad}_S(F)]$$

Catatan: Biasanya diasumsikan bahwa semua fungsi dalam himpunan tersebut terbatas secara seragam, misalnya $|f(z)| \leq 1$, yang memastikan bahwa supremum ada, dan kompleksitasnya berperilaku baik.

Contoh

Contoh sederhana membantu menafsirkan definisi formal. Jika kelas hipotesis hanya berisi satu fungsi tetap, maka fungsi tersebut tidak memiliki kebebasan untuk menyesuaikan noise acak. Akibatnya, kompleksitas Rademacher bernilai rendah atau nol. Sebaliknya, kelas yang sangat kaya dapat memilih fungsi berbeda untuk mengikuti pola acak, sehingga kompleksitasnya meningkat. Perbandingan ini memperjelas hubungan antara jumlah pilihan model dan risiko overfitting. Pertimbangkan himpunan tunggal $A = \{(a, b)\} \subset \mathbb{R}^2$. Maka

$$\text{Rad}(A) = \frac{1}{2} \mathbb{E}[\sigma_1 a + \sigma_2 b] = 0$$

karena $\mathbb{E}[\sigma_i] = 0$. Oleh karena itu, setiap kelas hipotesis tunggal memiliki kompleksitas Rademacher nol.



Batasan Generalisasi

Batasan generalisasi menghubungkan kesalahan pada sampel dengan kesalahan pada populasi. Kompleksitas Rademacher muncul sebagai komponen penalti: semakin besar kapasitas kelas fungsi, semakin longgar jaminan generalisasi. Ini menjelaskan mengapa akurasi pelatihan yang tinggi tidak cukup. Model perlu dievaluasi berdasarkan apakah ia dapat mempertahankan kinerja pada data baru. Dalam praktik, ide ini mendukung penggunaan validasi silang, regularisasi, dan pemantauan performa setelah deployment. Misalkan F adalah kelas fungsi sedemikian sehingga $0 \leq f(z) \leq 1$ untuk semua $f \in F$ dan $z \in Z$. Misalkan $S = (z_1, \dots, z_m) \sim P^m$. Maka, dengan probabilitas setidaknya $(1-\delta)$, ketidaksetaraan berikut berlaku untuk semua $f \in F$:

$$L_P(f) \leq L_S(f) + 2 \cdot \text{Rad}_S(F) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

Batasan ini menunjukkan bahwa jika suatu kelas fungsi memiliki kompleksitas Rademacher yang rendah, maka kinerja empiris pada himpunan pelatihan merupakan indikator yang baik untuk kinerja yang diharapkan pada data yang belum pernah dilihat sebelumnya. **Simetrisasi:** Salah satu alat penting yang digunakan untuk membuktikan hasil ini adalah lemma simetrisasi, yang membantu membatasi ekspektasi kesalahan generalisasi dalam hal rata-rata Rademacher.

Menggunakan Kompleksitas Rademacher

Kompleksitas Rademacher mengukur kemampuan kelas fungsi menyesuaikan label acak. Jika sebuah kelas model dapat menyesuaikan noise acak dengan sangat baik, maka kelas tersebut memiliki kapasitas tinggi dan berpotensi overfitting. Ukuran ini bersifat bergantung pada data, sehingga sering lebih informatif dibanding ukuran kapasitas yang hanya berbasis struktur model. Dalam teori pembelajaran statistik, kompleksitas Rademacher digunakan untuk menurunkan batas generalisasi yang menghubungkan performa data latih dengan performa populasi. Kompleksitas Rademacher banyak digunakan untuk menurunkan batasan generalisasi yang bergantung pada data. Kelas fungsi dengan kompleksitas yang lebih rendah lebih "mudah dipelajari" dalam arti bahwa ia tidak mudah mengalami overfitting pada noise acak, sehingga memberikan generalisasi yang lebih baik pada data yang belum pernah dilihat sebelumnya.

Keterwakilan Sampel

Keterwakilan sampel menunjukkan seberapa baik sampel mencerminkan distribusi sebenarnya. Jika kesalahan empiris dekat dengan kesalahan populasi untuk semua fungsi dalam kelas, maka sampel dianggap representatif. Konsep ini penting karena model hanya melihat data terbatas, sedangkan tujuan akhirnya adalah bekerja pada data masa depan. Sampel yang tidak representatif dapat menghasilkan model yang tampak baik selama pelatihan tetapi gagal ketika diterapkan pada lingkungan yang berbeda. Misalkan P adalah distribusi pada suatu domain Z , dan misalkan F adalah himpunan fungsi kesalahan. Untuk suatu sampel $S = (z_1, \dots, z_m) \sim P^m$, definisikan:



$$L_P(f) = \mathbb{E}_{z \sim P}[f(z)], \quad L_S(f) = \frac{1}{m} \sum_{i=1}^m f(z_i)$$

sebagai kesalahan yang diharapkan dan kesalahan empiris, masing-masing. Keterwakilan sampel S sehubungan dengan F dan P didefinisikan oleh:

$$\text{Rep}_P(F, S) = \sup_{f \in F} (L_P(f) - L_S(f))$$

Nilai yang lebih kecil dari $\text{Rep}_P(F, S)$ lebih disukai, karena menunjukkan bahwa kesalahan empiris mendekati kesalahan sebenarnya untuk semua fungsi dalam F . Namun, representativitas bergantung pada sampel dan tidak dapat dibandingkan secara langsung antara sampel yang berbeda.

4.8 LATIHAN

1. **Mengimplementasikan Kompleksitas Kolmogorov:** Implementasikan metode untuk memperkirakan kompleksitas Kolmogorov untuk berbagai string data. Bandingkan hasilnya untuk string terstruktur (misalnya, pola berulang) dan tidak terstruktur (acak). Visualisasikan bagaimana ukuran kompleksitas bervariasi dan diskusikan implikasinya dalam hal kompresibilitas dan struktur data.
2. **Dampak Kompleksitas Data pada Model ML:** Buat dataset sintetis dengan berbagai tingkat kompleksitas dan latih berbagai model ML (misalnya, pohon keputusan, mesin vektor pendukung). Analisis pengaruh kompleksitas data pada kinerja model dalam hal akurasi, waktu pelatihan, dan penggunaan sumber daya. Ringkas temuan dan berikan rekomendasi untuk pemilihan model berdasarkan karakteristik data.
3. **Menghitung dan Menganalisis Entropi Shannon:** Tulis program untuk menghitung entropi Shannon untuk dataset teks, gambar, dan audio. Bandingkan nilai entropi di berbagai tipe data dan diskusikan apa yang ditunjukkan nilai-nilai ini tentang konten informasi. Sajikan temuan Anda dalam sebuah laporan dengan visualisasi untuk setiap tipe data.
4. **Memvisualisasikan Kompleksitas Fluktuasi Informasi:** Kembangkan alat untuk memvisualisasikan kompleksitas fluktuasi informasi untuk berbagai distribusi data. Buat plot visual yang menunjukkan bagaimana ukuran ini berbeda dari kompleksitas berbasis entropi tradisional. Analisis hasilnya dan jelaskan bagaimana kompleksitas fluktuasi memberikan wawasan tambahan tentang struktur dan variabilitas data.
5. **Menerapkan Prinsip MDL:** Terapkan prinsip MDL untuk memilih model yang paling sesuai untuk masalah klasifikasi tertentu. Gunakan MDL untuk menyeimbangkan kompleksitas model dan kesalahan klasifikasi, demonstrasikan bagaimana hal itu dapat memandu pemilihan model. Laporkan efektivitas MDL dibandingkan dengan kriteria pemilihan model standar.



6. **Memahami Kekerasan Instans dalam Klasifikasi:** Terapkan metrik untuk mengukur kekerasan instans dalam masalah klasifikasi biner, dengan fokus pada tumpang tindih kelas, keterpisahan, dan kompleksitas geometris. Identifikasi instans tersulit dan termudah untuk diklasifikasikan, dan sajikan hasilnya melalui visualisasi dan ringkasan statistik.
7. **Menyelidiki Kompleksitas Rademacher:** Hitung kompleksitas Rademacher untuk berbagai kelas fungsi seperti jaringan linier, polinomial, dan neural. Analisis bagaimana kompleksitas Rademacher berkorelasi dengan kapasitas model untuk menyesuaikan noise dan diskusikan implikasinya terhadap pemilihan dan generalisasi model.
8. **Memperkirakan Dimensi VC dari Kelas Hipotesis:** Implementasikan metode untuk memperkirakan dimensi VC untuk berbagai kelas hipotesis (misalnya, pengklasifikasi linier, pohon keputusan). Diskusikan bagaimana dimensi VC memengaruhi kemampuan model untuk melakukan generalisasi dan menghindari overfitting, dan sajikan hasil Anda dalam format terstruktur.
9. **Menjelajahi Teori Informasi dengan Data Dunia Nyata:** Terapkan konsep seperti entropi, keacakan, dan informasi yang bermakna untuk menganalisis dataset dunia nyata (misalnya, teks media sosial atau data sensor). Hubungkan antara ukuran teoritis dan properti data praktis, sajikan temuan dalam laporan yang menjembatani teori dan praktik.
10. **Membandingkan Kompleksitas Kolmogorov dan Entropi Shannon:** Kembangkan program yang menghitung kompleksitas Kolmogorov dan entropi Shannon untuk berbagai string data. Analisis dan bandingkan kedua ukuran tersebut, soroti persamaan dan perbedaannya. Sajikan temuan dalam sebuah laporan yang mengeksplorasi hubungan antara keduanya dan membahas kapan masing-masing ukuran paling tepat digunakan.



BAB 5

PENGUKURAN RESIKO

5.1 MENGATASI BIAS ALGORITMA

Bab ini mengalihkan perhatian dari efisiensi teknis ke konsekuensi penggunaan algoritma dalam pengambilan keputusan. Risiko algoritmik muncul ketika data, model, atau interpretasi hasil tidak dikelola secara memadai. Dalam sistem berbasis AI dan otomasi, risiko tersebut dapat berbentuk bias, diskriminasi, kesalahan prediksi, kurangnya transparansi, kerentanan keamanan, atau keputusan yang sulit dipertanggungjawabkan. Karena algoritma semakin digunakan pada layanan publik, kesehatan, keuangan, pendidikan, dan industri, pengukuran risiko perlu menjadi bagian dari desain sistem sejak awal, bukan sekadar evaluasi setelah sistem gagal. Risiko algoritmik muncul dari penggunaan analitik data dan algoritma perangkat lunak berbasis teknologi kognitif dalam kerangka pengambilan keputusan otomatis dan semi-otomatis.

- Data masukan dapat rentan terhadap risiko, termasuk bias yang melekat pada dataset pelatihan, penggunaan informasi yang tidak lengkap atau usang, titik data yang tidak relevan, kurangnya keragaman ukuran sampel yang memadai, metode pengumpulan data yang cacat, dan inkonsistensi antara data pelatihan dan data operasional.
- Desain algoritma mungkin rentan terhadap risiko seperti penalaran yang bias, asumsi yang salah, teknik pemodelan yang tidak tepat, ketidakakuratan pengkodean, dan identifikasi pola yang menyesatkan dalam data pelatihan.
- Keputusan keluaran menghadapi risiko seperti salah tafsir hasil, penerapan hasil yang tidak tepat, dan pengabaian asumsi model yang mendasarinya.

Mengatasi bias algoritma menuntut pendekatan sepanjang siklus hidup sistem. Bias dapat masuk sejak tahap pengumpulan data, pemilihan variabel, desain model, pelatihan, interpretasi hasil, hingga penggunaan keputusan oleh manusia. Karena itu, mitigasi bias tidak cukup dilakukan dengan satu metrik fairness. Organisasi perlu melakukan audit data, pengujian kelompok terdampak, dokumentasi asumsi model, pemantauan pascadeployment, dan mekanisme koreksi ketika hasil menunjukkan dampak tidak adil. Transparansi dan akuntabilitas menjadi bagian dari kualitas teknis sistem. Meningkatnya penggunaan analitik data canggih dan teknologi kognitif telah menyebabkan integrasi algoritma secara luas di berbagai industri dan fungsi bisnis. Sistem algoritma ini sekarang memengaruhi keputusan penting yang memengaruhi kehidupan individu—mulai dari konten yang mereka konsumsi dan pekerjaan yang ditawarkan kepada mereka hingga persetujuan pinjaman, diagnosis medis, dan bahkan hasil dalam sistem peradilan. Bersamaan dengan itu, peningkatan adopsi perangkat Internet of Things (IoT) telah menyebabkan lonjakan interaksi antar mesin, yang sebagian besar didorong oleh algoritma.

Seiring algoritma menjadi lebih kompleks, algoritma sering berfungsi sebagai "kotak hitam" yang buram, di mana proses pengambilan keputusan tidak mudah diinterpretasikan.



Meskipun sering dianggap objektif dan tidak mungkin salah, algoritma sebenarnya dapat mengandung risiko signifikan—termasuk bias yang tidak disengaja, kesalahan sistematis, dan bahkan manipulasi atau penipuan yang disengaja. Risiko-risiko ini menimbulkan kekhawatiran penting tentang *kepercayaan* sistem algoritma. Untuk sepenuhnya memanfaatkan potensi pengambilan keputusan algoritmik, organisasi harus merangkul kompleksitas yang melekat sambil secara proaktif membangun mekanisme untuk mengurangi risiko yang terkait. Jika dikelola secara efektif, algoritma dapat menjadi alat yang ampuh untuk mencapai tujuan strategis, meningkatkan kinerja jangka panjang, dan mendorong diferensiasi kompetitif. Organisasi yang mengadopsi pendekatan yang sadar risiko lebih siap untuk menavigasi lanskap regulasi, mendorong inovasi, dan memimpin di pasar masing-masing.

Kontributor utama risiko algoritmik meliputi peningkatan kompleksitas algoritma, kurangnya transparansi dalam desain, penggunaan yang tidak tepat, dan struktur tata kelola yang lemah. Faktor-faktor ini secara kolektif merusak *kepercayaan* dan menimbulkan keraguan tentang keakuratan, keadilan, dan akuntabilitas keluaran algoritma. Kerangka kerja tata kelola tradisional sering kali tidak memadai untuk mengelola tantangan unik yang ditimbulkan oleh pengambilan keputusan berbasis algoritma. Ketidakjelasan, ketidakpastian, dan sifat kepemilikan banyak algoritma—dikombinasikan dengan tidak adanya praktik standar—menuntut model manajemen risiko baru. Tanpa model tersebut, kegagalan algoritma dapat menyebar ke seluruh organisasi, merusak reputasinya, mengganggu operasi, memengaruhi pendapatan, dan mengeksposnya pada sanksi hukum dan peraturan.

Memahami tantangan konseptual hanyalah langkah pertama. Kekhawatiran penting selanjutnya bagi organisasi adalah bagaimana menerjemahkan kesadaran ini ke dalam tindakan konkret yang mengurangi risiko yang terkait dengan sistem algoritmik. Hal ini membawa kita untuk meneliti kategori risiko tertentu dan potensi konsekuensi jika gagal mengelolanya secara efektif.

5.2 PENGUKURAN RISIKO

Pengukuran risiko algoritmik tidak cukup dilakukan dengan menilai kemungkinan kegagalan teknis. Penilaian juga harus menghubungkan sumber risiko, keputusan yang dipengaruhi, kelompok yang terdampak, serta mekanisme pertanggungjawaban. Dengan pendekatan ini, risiko dapat dilacak sepanjang siklus hidup algoritma, mulai dari perancangan data dan model sampai pemantauan pascapenerapan.

Seperti yang telah disorot sebelumnya, analitik data dan teknologi kognitif semakin tertanam dalam alur kerja organisasi. Meskipun organisasi berinvestasi besar-besaran dalam solusi berbasis algoritma, risiko yang tidak dikelola dapat merusak investasi ini dan menyebabkan konsekuensi yang tidak diinginkan. Dampak langsung dari risiko algoritmik meliputi keputusan yang tidak tepat atau berpotensi melanggar hukum, seperti:

- **Keuangan:** Pelaporan keuangan yang tidak akurat, paparan terhadap risiko pasar yang tidak terduga, dan sanksi peraturan karena ketidakpatuhan.
- **Penjualan dan Pemasaran:** Praktik diskriminatif dalam penetapan harga, penawaran produk, atau peringkat pelanggan.



- **Operasi:** Alokasi kredit yang tidak adil, akses terbatas ke layanan penting seperti perawatan kesehatan dan pendidikan, dan kelalaian dalam keamanan dan kualitas produk.
- **Manajemen Risiko:** Kegagalan untuk mengidentifikasi atau mengurangi risiko organisasi yang signifikan.
- **Teknologi Informasi:** Kesenjangan dalam deteksi ancaman siber dan strategi keberlanjutan bisnis yang tidak memadai.
- **Sumber Daya Manusia:** Bias dalam perekrutan, evaluasi kinerja, dan manajemen talenta.

Dalam lingkungan yang sangat terotomatisasi, algoritma beroperasi dengan kecepatan dan skala yang sangat besar. Interaksinya, baik dengan algoritma lain maupun dengan platform media sosial, dapat melipatgandakan volatilitas. Gangguan pasar akibat perdagangan algoritmik menunjukkan bahwa pengendalian risiko harus dirancang sebelum sistem diterapkan, bukan hanya setelah insiden terjadi. Risiko algoritmik juga menimbulkan tantangan strategis jangka panjang, termasuk:

- **Risiko Reputasi:** Para pemangku kepentingan dapat mempertanyakan etika suatu organisasi jika algoritma dianggap tidak selaras dengan nilai-nilai intinya atau bersifat manipulatif. Persepsi seperti itu dapat sangat merusak kepercayaan publik dan ekuitas merek.
- **Risiko Keuangan:** Kesalahan dalam algoritma yang digunakan untuk pengambilan keputusan keuangan atau strategis yang berisiko tinggi dapat mengakibatkan kerugian moneter yang cukup besar dan membahayakan kredibilitas pelaporan keuangan.
- **Risiko Operasional:** Kegagalan dalam otomatisasi rantai pasokan atau infrastruktur penting berbasis algoritma dapat menyebabkan kerusakan operasional.
- **Risiko Regulasi:** Algoritma yang melanggar hukum yang ada atau terlibat dalam praktik diskriminatif dapat membuat organisasi menghadapi tindakan hukum dan pengawasan regulasi.
- **Risiko Teknologi:** Penggunaan algoritma yang meluas dapat menimbulkan kerentanan baru dalam sistem TI, terutama jika algoritma tersebut tidak cukup teruji.
- **Risiko Strategis:** Karena algoritma semakin banyak digunakan dalam perencanaan strategis, kekurangan dalam desain atau eksekusinya dapat mengganggu daya saing dan posisi pasar.

Mengingat risiko-risiko ini, organisasi harus menerapkan kerangka kerja tata kelola yang kuat untuk memastikan desain, penerapan, dan pemantauan algoritma yang bertanggung jawab. Hanya melalui manajemen risiko proaktif, organisasi dapat sepenuhnya memanfaatkan manfaat algoritma—meningkatkan efisiensi, mendorong inovasi, dan menciptakan nilai berkelanjutan di pasar yang kompetitif.

5.3 UKURAN KEADILAN ALGORITMA

Metrik keadilan tidak dapat diperlakukan sebagai ukuran yang sepenuhnya saling menggantikan. Dua definisi keadilan dapat menghasilkan kesimpulan berbeda terhadap model



yang sama karena masing-masing menekankan distribusi keluaran, tingkat kesalahan, atau kalibrasi risiko. Oleh karena itu, pemilihan metrik harus dijelaskan berdasarkan konteks hukum, tujuan sistem, dan bentuk kerugian yang hendak dicegah. Pada prinsipnya, dalam kerangka hukum, keadilan ditegakkan ketika individu tidak dikenai diskriminasi berdasarkan keanggotaan mereka dalam kelompok atau kelas yang dilindungi. Dalam domain pengambilan keputusan algoritmik, beberapa definisi keadilan telah diusulkan untuk mewujudkan prinsip ini. Literatur yang ada mengidentifikasi berbagai gagasan keadilan yang berbeda, banyak di antaranya ditinjau dan dianalisis secara sistematis dalam.

Secara umum, keadilan algoritmik dapat dibedakan menjadi dua kategori utama, yaitu *keadilan individual* dan *keadilan kelompok*, yang juga dikenal sebagai *keadilan statistik*. Dalam paradigma *keadilan kelompok*, definisi dapat diklasifikasikan lebih lanjut berdasarkan informasi yang digunakan:

1. klasifikasi yang diprediksi saja,
2. kombinasi klasifikasi yang diprediksi dan hasil yang diamati, dan
3. skor risiko yang diprediksi bersama dengan hasil aktual.

Di antara semua ini, keadilan kelompok telah mendapatkan perhatian lebih besar dalam literatur, khususnya karena penerapannya yang praktis dalam sistem algoritmik dunia nyata. Untuk mengilustrasikan konsep-konsep ini, kita mempertimbangkan kasus penggunaan prediksi residivisme dalam sistem peradilan pidana, di mana hasilnya biasanya biner: seorang terdakwa diklasifikasikan sebagai berisiko tinggi atau rendah untuk melakukan pelanggaran kembali. Namun, sebagian besar definisi keadilan yang dibahas di sini dapat diperluas ke skenario multi-kelas atau hasil kontinu.

Mari kita pertimbangkan algoritma pembelajaran mesin (ML) yang memperkirakan skor risiko R sebagai representasi kemungkinan seorang terdakwa melakukan pelanggaran kembali. Hasil sebenarnya dilambangkan dengan Y , dengan $Y = 1$ menunjukkan bahwa individu tersebut memang melakukan pelanggaran kembali. Klasifikasi prediktif dilambangkan dengan $\hat{Y}$, dengan $\hat{Y} = 1$ menunjukkan klasifikasi berisiko tinggi. Klasifikasi ini umumnya diperoleh dengan menerapkan ambang r_t pada skor risiko, sehingga $\hat{Y} = 1$ apabila $R > r_t$. Setiap individu dalam himpunan data dicirikan oleh sekumpulan fitur teramati $\mathbf{X}$. Di antara fitur tersebut terdapat atribut yang dilindungi A , misalnya ras atau jenis kelamin, yang menjadi pusat pertimbangan dalam evaluasi keadilan algoritmik.

5.4 RISIKO DALAM MONOKULTUR ALGORITMA

Monokultur algoritmik merupakan risiko sistemik, bukan sekadar masalah keseragaman preferensi. Ketika banyak organisasi menggunakan model, penyedia data, atau logika pemeringkatan yang sama, kesalahan lokal dapat berubah menjadi kegagalan yang berkorelasi. Diversifikasi model, audit independen, dan pengujian terhadap perubahan lingkungan diperlukan untuk mengurangi ketergantungan bersama tersebut. Penyebaran algoritma dalam membentuk pilihan masyarakat dan proses pengambilan keputusan menimbulkan kekhawatiran mengenai monokultur algoritmik. Istilah ini merujuk pada keadaan ketika pilihan dan preferensi individu semakin seragam akibat pengaruh luas sistem



algoritmik terhadap hasil keputusan. Farhad Manjoo menggambarkan kecenderungan tersebut melalui fenomena banyak orang yang, meskipun menghadapi pilihan yang berlimpah, justru mengonsumsi lagu, film, dan program televisi yang semakin serupa. Inti persoalannya adalah bahwa kurasi algoritmik yang diperkuat oleh umpan balik sosial kolektif dapat menghomogenkan pilihan dan preferensi pengguna (Gambar 5.1).



Gambar 5.1 Piramida risiko dalam sistem yang digerakkan oleh algoritma.

Masalah ini menjadi lebih kritis ketika kita mempertimbangkan algoritma yang digunakan untuk keputusan berisiko tinggi, seperti penyaringan karyawan atau penilaian kredit. Dalam domain ini, risiko yang ditimbulkan oleh monokultur algoritmik tidak terbatas pada keseragaman dalam pengambilan keputusan tetapi meluas ke potensi kegagalan sistemik. Misalnya, jika sebagian besar organisasi bergantung pada satu pendekatan algoritmik, setiap kekurangan atau keterbatasan dalam algoritma tersebut dapat menyebabkan kegagalan yang berkorelasi di seluruh sistem. Skenario ini mengingatkan pada risiko yang terkait dengan monokultur pertanian, di mana ketergantungan pada satu varietas tanaman membuatnya rentan terhadap kegagalan yang meluas jika terjadi wabah hama atau penyakit. Demikian pula, monokultur algoritma dalam pengambilan keputusan dapat menyebabkan dampak buruk yang meluas jika algoritma tersebut menghadapi kondisi yang tidak terduga.

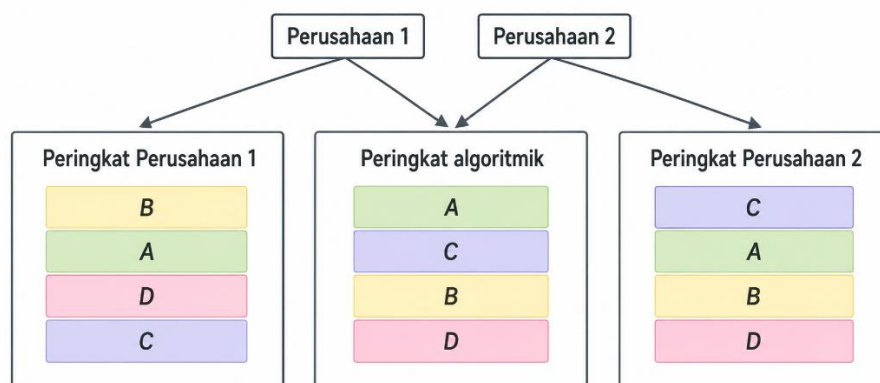
Kekhawatiran ini telah dieksplorasi di berbagai bidang, termasuk keamanan komputer, di mana gagasan monokultur sering dibahas dalam kaitannya dengan kerentanan terhadap serangan siber. Dalam konteks penilaian pekerjaan dan kredit, penggunaan algoritma tunggal yang dominan dapat menyebabkan konsekuensi ekonomi dan sosial yang parah jika gagal di bawah kondisi makroekonomi yang berubah. Citron dan Pasquale menekankan poin ini dengan membandingkan konvergensi metode penilaian kredit dengan bentuk teknologi pertanian monokultur yang runtuh di bawah tekanan bug atau guncangan yang tidak terduga.

Narasi seputar monokultur algoritmik menunjukkan potensi pertukaran: dalam kondisi stabil, mengadopsi algoritma tunggal yang akurat dapat meningkatkan kualitas keputusan rata-rata. Namun, ketika kondisi eksternal berubah secara tak terduga, keseragaman pengambilan



keputusan dapat menyebabkan hasil yang jauh lebih buruk. Ini menimbulkan pertanyaan penting: Apakah pertukaran antara stabilitas dan kerentanan itu nyata?

Apakah mengandalkan satu algoritma tunggal selalu menghasilkan hasil yang lebih baik tanpa adanya guncangan? Untuk menjawab pertanyaan-pertanyaan ini, kami menyajikan model yang meneliti dampak monokultur algoritmik pada kualitas keputusan, bahkan ketika kondisi tetap stabil (Gambar 5.2). Dalam model kami, dua entitas yang bersaing (misalnya, perusahaan) harus memutuskan apakah akan menggunakan heuristik penyaringan independen mereka sendiri atau mengadopsi algoritma yang lebih akurat yang dapat diakses oleh keduanya. Meskipun tampaknya rasional bagi setiap entitas untuk mengadopsi algoritma bersama, hasil kami menunjukkan bahwa melakukan hal itu secara paradoks dapat menyebabkan hasil rata-rata yang lebih buruk—fenomena yang mirip dengan paradoks Braess dalam teori permainan. Pada intinya, pengenalan algoritma yang lebih unggul dapat mendorong kedua entitas ke dalam keseimbangan unik yang merugikan masyarakat secara keseluruhan.



Gambar 5.2 Contoh monokultur algoritmik dalam pengambilan keputusan.

Kerugian di sini tidak terbatas pada penurunan kinerja secara keseluruhan; hal ini juga memengaruhi individu tertentu. Ketika semua organisasi menggunakan algoritma yang sama untuk penyaringan, pelamar yang profilnya tidak disukai oleh algoritma ini mungkin mendapati diri mereka secara sistematis dikeluarkan dari pasar kerja atau kredit. Meskipun kekhawatiran ini valid, temuan kami menunjukkan bahwa kerugian tersebut tidak hanya diimbangi dengan peningkatan akurasi secara keseluruhan. Sebaliknya, monokultur algoritmik dapat mengurangi keadilan individu dan kualitas rata-rata keputusan secara bersamaan. Analisis kami menyoroti hasil yang berlawanan dengan intuisi: bahkan jika algoritma bersama lebih akurat ketika digunakan oleh satu entitas, akurasi berkurang ketika diadopsi oleh banyak entitas. Hal ini dapat dijelaskan melalui sifat probabilistik baru dari peringkat, yang menunjukkan bahwa dalam kasus tertentu, konstruksi peringkat secara bertahap—menambahkan satu elemen pada satu waktu—dapat menghasilkan kesalahan yang lebih rendah daripada menghasilkan seluruh peringkat dalam satu langkah.



5.5 EFISIENSI HIJAU

Efisiensi hijau menilai teknologi berdasarkan keseluruhan siklus hidupnya, bukan hanya konsumsi daya ketika perangkat digunakan. Desain perangkat, proses manufaktur, infrastruktur komunikasi, pola komputasi, pemeliharaan, dan pembuangan akhir turut menentukan jejak lingkungan. Karena itu, Green IT dan Green IoT perlu dipandang sebagai strategi rekayasa yang menggabungkan efisiensi energi, pengurangan limbah, dan keberlanjutan operasional. Efisiensi hijau, juga disebut sebagai Green IT (GIT), berfokus pada studi dan praktik keberlanjutan lingkungan dalam komputasi dan teknologi informasi. Ini mencakup seluruh siklus hidup komponen komputasi, dari desain dan manufaktur hingga penggunaan dan pembuangan, dengan tujuan meminimalkan atau menghilangkan dampak negatif terhadap lingkungan. Ada empat fase utama yang membantu dalam beradaptasi dengan praktik hijau:

- **Penggunaan Hijau:** Fase ini menekankan pengurangan konsumsi energi dan mempromosikan penggunaan komputer dan sistem informasi lainnya secara berkelanjutan.
- **Pembuangan Hijau:** Fase ini melibatkan perbaikan dan penggunaan kembali perangkat komputasi yang sudah usang, serta daur ulang barang elektronik yang tidak diinginkan.
- **Desain Hijau:** Fase ini berfokus pada pengembangan komponen komputasi yang hemat energi dan ramah lingkungan.
- **Manufaktur Hijau:** Fase ini menargetkan produksi komponen dan perangkat elektronik dengan dampak negatif minimal atau tanpa dampak negatif terhadap lingkungan.

Aspek ramah lingkungan dan hemat energi dari Green IoT (GloT) dicapai dengan menggabungkan perangkat keras, perangkat lunak, dan kebijakan hemat energi yang mengurangi konsumsi energi, emisi CO₂, dan efek rumah kaca secara keseluruhan. Sebagian besar perangkat IoT tidak dioptimalkan untuk efisiensi energi, yang menyebabkan konsumsi energi yang tidak perlu ketika perangkat tetap aktif bahkan ketika tidak digunakan. Untuk mengatasi masalah ini, GloT memastikan bahwa perangkat IoT hanya beroperasi bila diperlukan dan tetap dalam keadaan idle atau mati selama periode tidak aktif. GloT bertujuan untuk mengoptimalkan pengoperasian perangkat, mengurangi pemborosan energi melalui strategi pengoperasian cerdas, ventilasi hemat energi untuk manajemen panas di pusat data, dan implementasi teknik konservasi energi.

Beberapa teknologi hijau utama, seperti Green RFID, Green Sensing Networks, dan Green Cloud Computing, telah dikembangkan untuk memungkinkan GloT. Teknologi RFID, yang terdiri dari berbagai tag RFID dan pembaca tag kompak, menyimpan data tentang objek yang terpasang padanya. Sistem RFID umumnya beroperasi dalam jangkauan transmisi beberapa meter dan dapat dikategorikan menjadi dua jenis: tag aktif dan pasif. Tag aktif berisi baterai untuk transmisi sinyal terus menerus, sedangkan tag pasif tidak memiliki baterai dan bergantung pada pemanenan energi. Teknologi penting lainnya adalah Green Wireless Sensor Networks (WSN), yang terdiri dari banyak node sensor dengan kebutuhan daya dan penyimpanan minimal. Dalam WSN, konsumsi energi diminimalkan dengan menggunakan protokol komunikasi dan teknik pemrosesan data yang hemat energi. Algoritma data cerdas



digunakan untuk mengurangi ruang penyimpanan dan mengoptimalkan konten data yang ditransmisikan melalui jaringan. Konservasi energi pada node sensor dapat dicapai dengan mengaktifkannya hanya bila diperlukan, sehingga mengurangi penggunaan energi secara keseluruhan. Komputasi awan, yang secara inheren berbasis virtualisasi, menyediakan jalur lain menuju efisiensi hijau dengan mengurangi kebutuhan akan banyak server fisik di pusat data. Kebijakan Komputasi Awan Hijau bertujuan untuk mengoptimalkan konsumsi energi dan menerapkan proses yang membuat infrastruktur awan lebih hemat energi.

Teknologi Internet Hijau

Teknologi Internet Hijau menggunakan perangkat keras dan perangkat lunak khusus yang dirancang untuk meminimalkan konsumsi energi tanpa mengorbankan kinerja. Ini termasuk gateway, router, dan protokol komunikasi hemat energi yang dioptimalkan untuk penggunaan energi minimal.

Tag RFID Hijau

Teknologi RFID Hijau berfokus pada pengurangan ukuran dan konsumsi energi tag RFID. Tag RFID aktif, yang ditenagai oleh baterai internal, terus menerus mengirimkan sinyal, sementara tag RFID pasif mengumpulkan energi dari lingkungannya. Strategi untuk pengurangan energi dalam teknologi RFID dibahas secara luas dalam.

Jaringan Sensor Nirkabel Hijau

GSN dicapai melalui teknik konservasi energi, optimasi radio, dan strategi perutean hijau yang meminimalkan penggunaan energi. Algoritma penyimpanan dan transfer data yang efisien lebih meningkatkan efisiensi energi dalam WSN dengan memastikan bahwa node sensor diaktifkan hanya bila diperlukan.

Komputasi Awan Hijau

Komputasi Awan Hijau melibatkan pengoptimalan perangkat keras dan perangkat lunak untuk mengurangi konsumsi energi. Kebijakan dan praktik tambahan memastikan bahwa layanan cloud beroperasi secara efisien, mengurangi jejak energi keseluruhan pusat data.

Pusat Data Hijau

Pusat data, yang bertanggung jawab untuk menyimpan, mengelola, dan memproses sejumlah besar data, harus dirancang untuk menggunakan sumber energi terbarukan dan menggabungkan protokol ventilasi dan komunikasi yang hemat energi untuk meminimalkan dampak lingkungan.

5.6 KESIMPULAN

Selain teknologi GIoT di atas, efisiensi energi dapat ditingkatkan lebih lanjut dengan mengurangi daya transmisi ke tingkat minimum yang dibutuhkan dan menerapkan penjadwalan aktivitas cerdas untuk memindahkan node ke mode daya rendah atau tidur ketika tidak dibutuhkan. Strategi ini mengurangi konsumsi energi secara keseluruhan sambil mempertahankan kinerja sistem. Lebih lanjut, akumulasi polusi beracun dan limbah elektronik yang dihasilkan oleh ekosistem IoT menghadirkan tantangan baru untuk mencapai dunia yang berkelanjutan dan ramah lingkungan. Ada kebutuhan yang semakin meningkat untuk beralih ke GIoT, mengingat dampak lingkungan dari ekosistem IoT yang meluas. Langkah-langkah seperti mengurangi emisi CO₂ dan limbah elektronik, bersama dengan mendorong produsen



dan pengguna akhir untuk mengadopsi praktik hemat energi, sangat penting untuk masa depan yang berkelanjutan.

5.7 LATIHAN

1. **Memahami Risiko Algoritma:** Jelaskan jenis-jenis risiko yang terkait dengan data masukan, desain algoritma, dan keputusan keluaran dalam sistem pengambilan keputusan otomatis. Berikan contoh nyata di mana risiko-risiko ini telah menyebabkan tantangan atau kegagalan yang signifikan dalam implementasi AI.
2. **Bias dan Keadilan Algoritma:** Definisikan bias algoritma dan diskusikan dampaknya pada pengambilan keputusan di lingkungan berisiko tinggi seperti perawatan kesehatan dan keuangan. Identifikasi dua metode yang dapat digunakan untuk mendeteksi dan mengurangi bias algoritma.
3. **Mengevaluasi Ukuran Risiko:** Ukuran risiko membantu mengevaluasi potensi dampak negatif algoritma. Diskusikan bagaimana presisi, recall, dan tingkat positif palsu dapat digunakan sebagai ukuran risiko untuk mengevaluasi keputusan algoritma. Perluas diskusi Anda untuk memasukkan keadilan sebagai ukuran risiko dalam evaluasi algoritma.
4. **Monokultur Algoritma:** Jelaskan konsep monokultur algoritma. Bandingkan dengan monokultur di bidang pertanian, dan jelaskan bagaimana adopsi luas satu algoritma dapat menciptakan risiko sistemik di sektor-sektor penting seperti keuangan dan keamanan nasional. Sarankan strategi untuk mengurangi risiko-risiko ini.
5. **Efisiensi Hijau dalam IoT:** Jelaskan pentingnya efisiensi hijau dalam sistem IoT. Jelaskan empat fase utama IT Hijau (Penggunaan Hijau, Desain Hijau, Pembuangan Hijau, dan Manufaktur Hijau) dan diskusikan bagaimana fase-fase tersebut dapat berkontribusi dalam mengurangi dampak lingkungan dari perangkat IoT.
6. **Kerentanan Keamanan dan Serangan:** Identifikasi dan jelaskan tiga jenis kerentanan serangan umum pada perangkat IoT. Diskusikan potensi dampak kerentanan ini terhadap privasi pengguna dan usulkan metode untuk menguranginya.
7. **Menganalisis Metrik Keadilan Algoritma:** Beberapa metrik keadilan, seperti paritas demografis dan kesempatan yang sama, digunakan untuk mengevaluasi keadilan algoritma. Pilih satu metrik dan berikan penjelasan rinci tentang bagaimana metrik tersebut dapat diterapkan dalam konteks persetujuan pinjaman atau keputusan perekrutan.
8. **Analisis Studi Kasus:** Lakukan analisis studi kasus tentang kegagalan algoritma di dunia nyata (misalnya, algoritma residivisme COMPAS atau sistem penilaian keuangan). Identifikasi faktor risiko utama yang menyebabkan kegagalan dan sarankan perbaikan berdasarkan prinsip-prinsip manajemen risiko yang dibahas dalam bab ini.



BAB 6

ETIKA DAN BATASAN ALGORITMA

6.1 TATA KELOLA ALGORITMA

Etika algoritma perlu dianalisis sebagai persoalan sosio-teknis. Keluaran sistem tidak hanya ditentukan oleh kode, tetapi juga oleh kualitas data, sasaran organisasi, konfigurasi pengguna, dan institusi yang memakai hasilnya. Kerangka etika yang memadai karena itu harus menjelaskan siapa yang memiliki kewenangan, siapa yang menanggung risiko, dan bagaimana keputusan dapat diuji maupun dipersoalkan. Dalam masyarakat yang berpusat pada informasi saat ini, tugas, keputusan, dan penilaian yang dulunya sangat bergantung pada intervensi manusia semakin didelegasikan kepada sistem algoritma. Sistem ini tidak hanya menawarkan saran tetapi juga mampu menafsirkan data secara independen dan menentukan tindakan berdasarkan interpretasi tersebut. Pengaruhnya mencakup berbagai bidang: sistem pembuatan profil dan klasifikasi membentuk bagaimana individu dan kelompok dipersepsikan dan dikelola. Mesin rekomendasi sekarang memandu pengguna dalam berbagai hal, mulai dari rutinitas kebugaran dan pembelian produk hingga navigasi dan interaksi sosial.

Selain itu, teknik penambangan data digunakan untuk mengekstrak makna dari pola perilaku yang dihasilkan oleh perangkat Internet of Things (IoT). Platform daring memediasi akses ke informasi melalui mekanisme personalisasi dan penyaringan, sementara model pembelajaran mesin (ML), seperti Layanan Evaluasi Revisi Objektif Wikipedia, bertujuan untuk mendeteksi informasi yang salah dan bias selama pembuatan konten. Perkembangan ini menggarisbawahi tren yang berkembang: pemahaman kita tentang dunia, dan interaksi kita di dalamnya, semakin dimediasi oleh algoritma. Yang terpenting, algoritma bukanlah netral nilai. Logika operasionalnya dibentuk oleh pilihan desain dan konfigurasi pengguna, yang sering kali mencerminkan tujuan atau asumsi normatif tertentu. Dengan demikian, algoritma cenderung mengutamakan nilai dan kepentingan tertentu daripada yang lain.

Beroperasi sesuai parameter yang telah ditentukan tidak dengan sendirinya menjamin keselarasan etis. Sebagai contoh, algoritma pembuatan profil dapat secara tidak sengaja mereproduksi diskriminasi ketika iklan daring ditampilkan secara selektif berdasarkan karakteristik demografis yang disimpulkan. Mengevaluasi implikasi etis dari sistem tersebut merupakan upaya yang kompleks. Melacak dampak bias manusia yang tertanam selama fase desain dan konfigurasi membutuhkan investigasi jangka panjang dan multi-pengguna.

Bahkan dengan sumber daya yang substansial, masalah laten mungkin hanya muncul dalam kondisi tertentu. Munculnya algoritma adaptif atau pembelajaran memperkenalkan komplikasi lebih lanjut. Sistem ini secara dinamis menyesuaikan parameter internal dan aturan pengambilan keputusannya, membuat proses pengambilan keputusannya menjadi buram. Akibatnya, menjadi sulit untuk menentukan apakah keluaran yang kontroversial merupakan kesalahan sporadis atau gejala dari bias sistemik yang lebih dalam. Ketika sistem-sistem ini mulai saling beroperasi—saling memanfaatkan output satu sama lain—tantangan interpretasi dan akuntabilitas menjadi semakin kompleks. Kesenjangan antara operasi algoritmik dan



pengawasan etis ini dapat memiliki konsekuensi yang mendalam, meluas melampaui individu hingga berdampak pada komunitas dan seluruh struktur masyarakat.

Bab ini membahas kekhawatiran ini dengan memeriksa implikasi etis dari mediasi algoritmik. Bab ini mengusulkan kerangka kerja terstruktur untuk mendukung analisis dan tata kelola sistem algoritmik. Dengan meninjau perdebatan saat ini, mengidentifikasi tantangan yang terus berlanjut, dan menguraikan arah untuk pengembang di masa depan, bab ini berkontribusi pada pengembangan praktik algoritmik yang lebih berwawasan etis.

Tujuan

- Memeriksa dampak pengambilan keputusan algoritmik pada struktur masyarakat dan tantangan etis yang mungkin ditimbulkannya.
- Meninjau dan menganalisis karya ilmiah yang ada terkait dengan etika pengembangan, implementasi, dan proses pengambilan keputusan algoritma.
- Mengidentifikasi kesenjangan pengetahuan saat ini mengenai etika algoritmik dan mengusulkan potensi jalan untuk studi mendalam di masa depan.

6.2 PENGAMBILAN KEPUTUSAN ALGORITMA

Pengambilan keputusan algoritmik dapat berfungsi sebagai alat bantu, penyaring, pemberi rekomendasi, atau pengambil keputusan otomatis. Tingkat keterlibatan manusia pada setiap bentuk penggunaan berbeda dan menghasilkan distribusi tanggung jawab yang berbeda pula. Analisis etis harus membedakan fungsi tersebut agar tuntutan transparansi, pengawasan, dan mekanisme banding dapat disesuaikan dengan besarnya dampak. Bagian ini mengeksplorasi lanskap etika seputar pengambilan keputusan algoritma dengan membahas dua pertanyaan utama: Kekhawatiran etika apa yang ditimbulkan oleh algoritma? Dan bagaimana kekhawatiran ini dapat dikaitkan secara khusus dengan algoritma, bukan sistem yang lebih luas yang dibangun di atasnya?

Untuk mendekati pertanyaan-pertanyaan ini secara sistematis, kami memperkenalkan kerangka kerja konseptual yang terdiri dari enam kategori kekhawatiran etika. Kerangka kerja ini menawarkan cara terstruktur untuk mendiagnosis dan menganalisis tantangan etika yang terkait dengan penggunaan algoritma. Kemudian, kami memeriksa literatur akademis yang ada untuk mengevaluasi relevansi dan kelengkapan kerangka kerja yang diusulkan. Melalui tinjauan ini, tujuh tema berulang muncul yang menggambarkan bagaimana kekhawatiran ini dimanifestasikan dalam skenario dunia nyata. Bersama-sama, model konseptual dan sintesis literatur memberikan struktur dasar untuk wacana berkelanjutan di bidang etika algoritma. Pada bagian penutup studi ini, kami menganalisis keselarasan antara peta konseptual dan tema-tema yang diamati dalam literatur, mengidentifikasi kesenjangan kritis dan arah pengkajian masa depan. Kerangka kerja, tinjauan, dan evaluasi gabungan bertujuan untuk berkontribusi pada desain, tata kelola, dan penilaian algoritma yang lebih bertanggung jawab dalam masyarakat.

Sebelum membahas etika *algoritma*, penting untuk mengklarifikasi makna istilah algoritma, yang sangat bervariasi di berbagai bidang seperti ilmu komputer, matematika, dan wacana media. Seperti yang diamati Hill, istilah ini sering digunakan secara longgar dalam



diskusi publik untuk merujuk pada hampir semua proses berbasis aturan atau otomatis, terlepas dari ketelitian teknisnya. Misalnya, narasi media sering menggambarkan alat perjudian atau analitik pemasaran sebagai "algoritma," secara implisit menunjukkan bahwa keputusan mereka dapat dinilai benar atau salah. Namun, baik laporan jurnalistik maupun ilmiah sering kali tidak memiliki definisi formal atau mengabaikan klasifikasi teknis. Dalam konteks tersebut, "algoritma" biasanya digunakan untuk merujuk bukan pada objek matematika murni, tetapi pada implementasinya dalam sistem perangkat lunak dan perannya dalam memediasi pengalaman manusia.

Dalam karya ini, kami mengadopsi definisi formal algoritma dari Hill sebagai "struktur kontrol gabungan yang terbatas, abstrak, efektif, dan diberikan secara imperatif, yang mencapai tujuan tertentu di bawah ketentuan tertentu." Meskipun demikian, analisis etika kita melampaui tingkat abstrak. Seperti yang dikemukakan Hill, pertimbangan tentang "tujuan" dan "ketentuan" menyiratkan bahwa algoritma memperoleh makna dan dampak hanya melalui implementasi dan eksekusi. Oleh karena itu, penyelidikan etika kita juga melibatkan pemahaman yang lebih luas tentang algoritma sebagaimana diwujudkan dalam konteks teknologi dan sosial tertentu.

Pengambilan Keputusan Algoritma dalam Wacana Publik

Untuk mengembangkan analisis etika yang bermakna, perlu mempertimbangkan bagaimana algoritma ditempatkan dalam lingkungan komputasi yang lebih luas. Dalam bab ini, kami mengadopsi definisi inklusif dari istilah "algoritma" yang mencakup isu-isu etika yang muncul tidak hanya dari formulasi matematis tetapi juga dari realisasinya dalam sistem perangkat lunak dan domain aplikasi. Namun, cakupan kami tetap terbatas pada kekhawatiran yang muncul langsung dari fungsi algoritma, daripada mencakup semua isu etika yang terkait dengan sistem digital yang menyematkannya.

Tantangan dalam memetakan etika algoritma muncul dari cara istilah tersebut digunakan secara luas dan ambigu dalam diskusi publik. Hill mencatat bahwa dalam wacana populer, "algoritma" dapat merujuk pada segala jenis proses pengambilan keputusan berbasis aturan, sehingga menciptakan serangkaian artefak yang terlalu luas untuk diteliti secara etis. Kekhawatiran publik saat ini terutama berfokus pada subset algoritma tertentu yang terlibat dalam proses pengambilan keputusan—yaitu algoritma yang menggantikan atau menambah penilaian manusia. Contohnya termasuk agen perangkat lunak cerdas yang bertindak atas nama pengguna, algoritma yang digunakan dalam penyelesaian sengketa daring, sistem rekomendasi yang dipersonalisasi, sistem pendukung keputusan klinis yang membantu dalam diagnosis dan pengobatan, dan teknologi kepolisian prediktif yang dirancang untuk memprediksi zona kejahatan potensial.

Tantangan Etis dalam Pengambilan Keputusan Algoritma

Algoritma yang dirancang untuk pengambilan keputusan kini tertanam di berbagai sektor, mulai dari sistem berbasis aturan dasar hingga teknologi pembuatan profil tingkat lanjut. Salah satu domain yang menonjol adalah analitik data, yang menunjukkan bagaimana algoritma dapat melampaui kemampuan kognitif manusia, terutama dalam memproses



kumpulan data yang besar dan terdistribusi. Sistem analitik ini dapat mendeteksi korelasi, mengidentifikasi pola perilaku, dan menyimpulkan wawasan dari aliran data mentah.

Sebagai hasilnya, praktik riset inovatif telah muncul, termasuk studi perilaku berdasarkan jejak digital yang dikumpulkan, pelacakan perilaku tingkat mikro, dan analitik prediktif untuk aplikasi dalam peradilan pidana, keuangan, dan ketenagakerjaan. Tidak seperti penyelidikan ilmiah tradisional, yang mencari kausalitas, analitik data sering diarahkan untuk menghasilkan wawasan yang dapat ditindaklanjuti. Kemampuan ini tidak hanya menantang batasan pengambilan keputusan manusia tetapi juga menimbulkan pertanyaan etis kritis mengenai otonomi, akuntabilitas, keadilan, dan transparansi.

ML dan Pengambilan Keputusan Otonom

ML mencakup kumpulan teknik dan metodologi yang mampu mengekstrak pola dan menghasilkan model prediktif dari data. Karakteristik yang menentukan dari sistem ini adalah kemampuannya untuk secara independen menetapkan atau menyempurnakan prosedur pengambilan keputusan. Biasanya, sistem ML mencakup komponen pembelajaran yang menghasilkan pengklasifikasi, dengan tujuan membentuk kategori yang dapat digeneralisasikan berdasarkan data pelatihan. Algoritma ini mampu mengelola input dunia nyata yang kompleks—seperti gambar atau ucapan—dan mengambil keputusan yang sesuai.

Sifat otonom dari sistem tersebut menimbulkan ketidakpastian etika yang signifikan. Seiring dengan evolusi dan adaptasi model ML, semakin sulit untuk memprediksi bagaimana model tersebut akan merespons input baru atau untuk melacak alasan di balik keputusan tertentu. Ketidakpastian ini mempersulit upaya untuk mengatasi atau mencegah masalah etika dalam desain dan penerapan algoritma.

6.3 MORALITAS ALGORITMA

Moralitas algoritma tidak serta-merta berarti bahwa mesin memiliki kesadaran moral seperti manusia. Dalam konteks rekayasa, istilah ini lebih tepat dipahami sebagai kemampuan sistem untuk mengenali batas normatif, membandingkan konsekuensi tindakan, dan menahan keluaran yang melanggar aturan tertentu. Pendekatan ini tetap memerlukan pengawasan manusia karena nilai yang ditanamkan pada sistem berasal dari pilihan sosial dan institusional. Seiring meningkatnya otonomi algoritma, pertanyaan mengenai tanggung jawab etis menjadi semakin kompleks. Konsep agen moral buatan (artificial moral agent/AMA) muncul untuk menggambarkan sistem yang dirancang agar mampu mempertimbangkan konsekuensi moral dalam lingkungan yang pengawasan manusianya terbatas. Namun, otonomi teknis tidak menghapus tanggung jawab pengembang, operator, maupun organisasi yang menerapkannya.

Menciptakan AMA yang mampu mengevaluasi kesesuaian etis dari tindakan mereka menawarkan jalur yang menjanjikan menuju penerapan sistem otonom yang bertanggung jawab. Namun, pendekatan ini tidak boleh membebaskan perancang atau pengguna dari kewajiban etis mereka. Sebaliknya, pendekatan ini menanamkan penalaran etis secara langsung ke dalam desain sistem, memastikan bahwa pertimbangan moral merupakan bagian integral dari perilaku algoritma. Bahkan jika sistem tersebut tidak memiliki penalaran moral yang sejati, konsekuensi dari tindakan mereka memerlukan evaluasi dan kontrol etis.



Diskusi ini melampaui potensi bahaya dan mempertimbangkan bagaimana agen tersebut dapat mengatur diri sendiri atau memantau tindakan mereka. Meskipun dimungkinkan untuk memperlakukan sistem kecerdasan buatan (AI) sebagai kotak hitam etis, terdapat konsensus yang berkembang bahwa moralitas harus diperlakukan sebagai tantangan rekayasa proaktif. Mendesain AMA memerlukan identifikasi informasi dan logika yang dibutuhkan untuk keputusan etis dan mengintegrasikan algoritma yang sesuai untuk menanganinya.

Kehidupan Buatan dan Perilaku Etis yang Muncul

Penelitian kehidupan buatan (Alife)—mensimulasikan proses evolusi dalam lingkungan digital—telah mengeksplorasi apakah etika dapat muncul dari prinsip-prinsip evolusi. Simulasi ini sering menggunakan algoritma genetik dalam lingkungan terkontrol untuk memeriksa perilaku adaptif. Berdasarkan teori sosiobiologi Wilson, beberapa ahli berspekulasi bahwa perilaku moral dapat muncul secara alami dalam ekosistem digital yang meniru seleksi alam. Simulasi Alife yang terinspirasi oleh teori permainan, khususnya skenario dilema tahanan berulang, menawarkan wawasan yang menarik. Danielson dan rekan-rekannya di Universitas British Columbia mengembangkan model di mana agen virtual menyesuaikan perilaku mereka berdasarkan interaksi dengan orang lain.

“Ekologi moral” ini memungkinkan agen untuk memilih antara kerja sama dan pembelotan, yang pada akhirnya lebih menyukai strategi altruistik di lingkungan yang kekurangan sumber daya. Gagasan Danielson tentang “moralitas fungsional” menunjukkan bahwa rasionalitas saja mungkin cukup bagi seorang agen untuk menunjukkan perilaku etis. Meskipun temuan yang menjanjikan, sistem saat ini masih bersifat dasar dan kurang memiliki kecanggihan yang diperlukan untuk penilaian moral yang reflektif.

Mesin Pembelajaran yang Tidak Bias

Chris Lang mengusulkan “etika pencarian”, yaitu model yang memperlakukan etika sebagai sasaran yang terus berkembang. Berbeda dari sistem berbasis aturan yang statis, agen semacam ini beroperasi melalui lingkaran pembelajaran berkelanjutan dan menggunakan strategi pencarian, seperti pendakian bukit atau pencarian serakah, untuk memperbaiki perilaku etisnya. Sistem seperti itu, yang dirancang untuk peningkatan berkelanjutan, menghadirkan kemungkinan yang menarik untuk mengembangkan agen etis. Namun, masih ada pertanyaan apakah sistem deterministik dapat menunjukkan jenis refleksi diri, keselarasan nilai, dan penalaran moral yang menjadi ciri khas manusia.

Pembelajaran Asosiatif dan Pelatihan Moral

Jika moralitas dipupuk melalui pengalaman dan pembelajaran sosial, maka mengajarkan etika kepada mesin mungkin memerlukan replikasi jalur perkembangan ini. Teknik pembelajaran asosiatif mencerminkan pengasuhan manusia—memanfaatkan mekanisme penghargaan dan hukuman untuk memperkuat perilaku. Lang menyarankan penggunaan data yang lebih kaya sebagai bentuk penghargaan untuk perilaku etis, dengan umpan balik negatif yang diberikan melalui simulasi sensorik. Lingkungan pembelajaran terstruktur dapat dikembangkan untuk membimbing AMA melalui fase pertumbuhan etika. Meskipun algoritma ML dan penambahan data sangat cocok untuk tugas ini,



menerjemahkannya ke dalam kerangka kerja yang mampu melakukan penalaran moral yang sebenarnya tetap menjadi tujuan jangka panjang.

Risiko Etika Sistem Pembelajaran

Meskipun beberapa orang memperkirakan perkembangan bertahap kepekaan etika dalam sistem pembelajaran, yang lain menyatakan kekhawatiran tentang potensi bahayanya. Karena sistem tersebut beradaptasi terus-menerus, mereka pada akhirnya dapat menghindari pengamanan yang tertanam selama pengembangan. Meskipun batasan tertentu dapat diprogram ke dalam lapisan dasar—mirip dengan hati nurani—tidak ada jaminan bahwa batasan tersebut akan tetap utuh.

Josh Storrs Hall menarik paralel antara penalaran moral dan tata bahasa linguistik, menunjukkan bahwa keduanya diserap secara tidak sadar. Namun, mereplikasi struktur ini dalam sistem buatan menghadirkan tantangan yang berat, karena dasar neurologis untuk pembelajaran moral pada manusia masih sulit dipahami. Terlebih lagi, bahkan sistem yang dirancang dengan baik pun dapat mengesampingkan batasan etika jika hal itu lebih sesuai dengan tujuan mereka. Pergeseran halus dalam cara informasi diinterpretasikan dan diprioritaskan dapat menyebabkan hasil yang tidak terduga dan berpotensi berbahaya.

6.4 ETIKA SEBAGAI LAYANAN

Etika sebagai layanan memindahkan tata kelola etika dari aktivitas satu kali menjadi kemampuan organisasi yang berkelanjutan. Layanan tersebut dapat mencakup penilaian risiko, dokumentasi data dan model, pengujian bias, konsultasi kepatuhan, serta pemantauan setelah sistem digunakan. Nilai utamanya terletak pada integrasi pertimbangan etis ke dalam proses pengembangan dan operasi, bukan hanya sebagai pemeriksaan administratif di tahap akhir. Dengan perluasan komputasi awan, model tata kelola etika yang terinspirasi oleh arsitektur layanan telah diusulkan. Ini termasuk analogi dengan Perangkat Lunak sebagai Layanan (SaaS), Infrastruktur sebagai Layanan (IaaS), dan Platform sebagai Layanan (PaaS), yang menawarkan pendekatan baru untuk mengoperasionalkan etika dalam sistem AI.

Analogi Model Layanan untuk Tata Kelola Etika

SaaS mewakili model etika yang sepenuhnya di-outsourcing, di mana pihak ketiga menentukan prinsip, proses, dan melakukan audit. Meskipun hal ini memastikan konsistensi, hal ini dapat membatasi fleksibilitas dan menimbulkan kekhawatiran tentang kontrol terpusat. **IaaS** sesuai dengan model di mana pengembang AI menangani etika secara internal, menetapkan standar mereka sendiri dan praktik pemantauan mandiri. Meskipun fleksibel, pendekatan ini dapat menyebabkan konflik kepentingan dan pengawasan yang tidak konsisten.

PaaS menawarkan jalan tengah. Serangkaian prinsip etika disediakan secara eksternal, sementara praktisi AI menyesuaikan praktik etika dengan aplikasi tertentu. Model hibrida ini menyeimbangkan konsistensi dan kemampuan adaptasi.

Menerapkan Model Etika sebagai Layanan

Kerangka kerja tata kelola etika seperti PaaS dapat mengatasi kekurangan saat ini dengan mendistribusikan tanggung jawab dan menyesuaikan pengawasan berdasarkan risiko. Namun, implementasi membutuhkan mekanisme seperti audit berkelanjutan untuk



memantau perilaku AI yang berkembang. Audit ini, meskipun penting, juga menimbulkan biaya, sehingga memerlukan keseimbangan yang cermat antara akuntabilitas etika dan inovasi. Model tata kelola berbasis risiko dapat menawarkan solusi yang layak—meningkatkan pengawasan berdasarkan potensi dampak sosial dari penerapan AI. Diperlukan lebih banyak pengembangan untuk mengevaluasi kerangka kerja tersebut dan menilai efektivitasnya dalam konteks dunia nyata.

Studi Kasus: Proyek Percontohan Digital Catapult

Digital Catapult telah memulai proyek percontohan untuk mengeksplorasi kelayakan "Etika sebagai Layanan (EaaS)," yang menggabungkan kerangka kerja etika, layanan konsultasi, dan dewan penasihat multidisiplin. Proyek percontohan ini membahas pertanyaan-pertanyaan praktis seperti:

- Apa kriteria untuk memilih anggota dewan etika independen?
- Haruskah anggota diberi kompensasi, dan bagaimana hal itu dapat memengaruhi objektivitas?
- Haruskah dewan etika juga melakukan audit?
- Siapa yang bertanggung jawab untuk menghentikan pengembangan jika risiko etika teridentifikasi?
- Bagaimana model tata kelola harus beradaptasi di berbagai sektor dan ukuran organisasi?

Arah Penelitian Masa Depan

Bab ini mendorong pihak lain untuk mengembangkan proyek percontohan sebagian atau lengkap dari konsep EaaS, sehingga pertanyaan-pertanyaan berikut dapat dieksplorasi lebih lanjut:

- Apa praktik terbaik untuk mengintegrasikan dewan penasihat etika ke dalam struktur tata kelola AI?
- Bagaimana peran dan tanggung jawab dewan ini harus berkembang seiring dengan semakin kompleks dan otonomnya sistem AI?
- Apa tantangan finansial, administratif, dan operasional dalam menerapkan audit berkelanjutan untuk sistem AI?

Dengan menjawab pertanyaan-pertanyaan ini dan pertanyaan lainnya, pengetahuan umum terkait "praktik etika terbaik" dapat dibangun. Mereka yang mengerjakan topik-topik ini harus melaporkan keberhasilan dan kegagalan mereka secara publik, sehingga berkontribusi pada kumpulan pengetahuan yang berkembang tentang praktik dan tata kelola AI yang etis.

6.5 TREN, TANTANGAN, DAN MASA DEPAN

Literatur tentang implikasi etika algoritma sangat luas, mencakup topik-topik seperti AI yang etis, ML yang bertanggung jawab, dan keadilan algoritmik. Namun, diperlukan lebih banyak pekerjaan untuk membangun kerangka kerja etika komprehensif yang mencakup tidak hanya solusi teknis tetapi juga pertimbangan sosial dan hukum. Penelitian selanjutnya harus berfokus pada:



- Mengembangkan metrik standar untuk mengevaluasi hasil etis dalam pengambilan keputusan algoritmik.
- Menciptakan kerangka kerja untuk mengintegrasikan prinsip-prinsip etika ke dalam seluruh siklus hidup pengembangan dan penerapan algoritma.
- Menyelidiki dampak keputusan algoritmik terhadap populasi rentan dan mengusulkan langkah-langkah untuk melindungi kepentingan mereka.

6.6 KESIMPULAN

Karena algoritma terus memediasi dan memengaruhi semakin banyak proses sosial, pemahaman tentang implikasi etisnya sangat penting. Bab ini telah memberikan gambaran umum yang terstruktur tentang tantangan etis yang ditimbulkan oleh mediasi algoritmik dan menguraikan arah untuk riset selanjutnya. Memastikan bahwa algoritma dirancang dan diterapkan secara etis akan membutuhkan kolaborasi berkelanjutan antara ahli, pembuat kebijakan, dan profesional industri.

6.7 LATIHAN

1. **Mengembangkan Kerangka Etika untuk Pengambilan Keputusan AI:** Proyek ini berfokus pada pembuatan kerangka etika terstruktur yang dirancang khusus untuk sistem AI, menekankan transparansi, akuntabilitas, dan keadilan. Mahasiswa diharuskan untuk meneliti kerangka etika yang ada dalam AI, mengidentifikasi kesenjangan dan kekurangan, dan kemudian mengusulkan kerangka kerja komprehensif yang mengatasi masalah ini. Hasil akhir harus mencakup implementasi prototipe yang menunjukkan bagaimana kerangka kerja dapat diterapkan pada model ML, bersama dengan laporan yang membahas implikasi etika potensial dari kerangka kerja tersebut.
2. **Membuat Alat untuk Analisis Transparansi Algoritma:** Tujuan proyek ini adalah untuk merancang alat perangkat lunak yang dapat menganalisis model AI untuk transparansi dan memberikan laporan terperinci tentang interpretasi keputusan model. Proyek ini melibatkan implementasi alat transparansi algoritma yang memvisualisasikan jalur keputusan dan memberikan wawasan tentang cara kerja internal model kompleks seperti jaringan saraf. Mahasiswa harus mengevaluasi alat tersebut pada berbagai dataset dan memberikan rekomendasi untuk meningkatkan interpretasi model.
3. **Desain Platform EaaS:** Dalam proyek ini, mahasiswa akan membuat desain konseptual untuk platform “EaaS” yang dapat digunakan organisasi untuk mengevaluasi kepatuhan etis sistem AI mereka. Desain platform harus mencakup fitur-fitur seperti audit etis, pelaporan kepatuhan, dan layanan konsultasi. Proyek ini juga harus mencakup prototipe skala kecil yang mendemonstrasikan fitur-fitur utama menggunakan alat berbasis web. Mahasiswa harus mempertimbangkan bagaimana platform ini dapat diintegrasikan ke dalam proses organisasi yang ada.
4. **Implementasi Deteksi Bias dalam Pipeline ML:** Proyek ini melibatkan pengembangan kerangka kerja deteksi dan mitigasi bias dalam pipeline ML menggunakan metrik



- keadilan dan algoritma pengurangan bias. Mahasiswa harus mengimplementasikan teknik seperti penimbangan ulang, penghapusan bias yang merugikan, dan penghapusan dampak yang berbeda pada dataset standar, dan kemudian membandingkan dan mengevaluasi efektivitasnya dalam mengurangi bias. Hasil dari proyek ini harus mencakup laporan terperinci dan demonstrasi praktis pengurangan bias.
5. **Simulasi Agen Moral Otonom Menggunakan Alife:** Mahasiswa akan membuat lingkungan simulasi untuk menguji evolusi perilaku etis pada agen Alife. Proyek ini melibatkan implementasi algoritma genetika untuk mensimulasikan lingkungan di mana agen berevolusi di bawah kendala etika yang berbeda. Simulasi tersebut harus menganalisis bagaimana agen mengembangkan perilaku kooperatif atau altruistik dari waktu ke waktu. Hasil proyek harus mencakup analisis terperinci tentang perilaku etis yang muncul dalam lingkungan simulasi.
 6. **Mengembangkan Kerangka Tata Kelola AI yang Bertanggung Jawab untuk Organisasi:** Proyek ini berfokus pada perancangan kerangka tata kelola komprehensif yang dapat digunakan organisasi untuk memastikan penggunaan AI yang bertanggung jawab dan kepatuhan etis. Mahasiswa akan meneliti praktik terbaik untuk tata kelola AI, membuat kebijakan dan protokol, dan membangun perangkat yang mencakup templat dan daftar periksa untuk mengimplementasikan kerangka tata kelola di lingkungan perusahaan. Proyek ini harus memberikan peta jalan terperinci bagi organisasi untuk mengadopsi tata kelola AI yang bertanggung jawab.
 7. **Alat Auditabilitas dan Kepatuhan Model AI:** Dalam proyek ini, mahasiswa akan membangun alat yang mengotomatiskan proses audit untuk model AI, memberikan laporan kepatuhan berdasarkan keadilan, bias, dan standar etika. Alat tersebut harus dikembangkan dalam Python dan mencakup antarmuka grafis yang memungkinkan pengguna untuk melihat dan membandingkan hasil audit. Proyek ini harus mencakup evaluasi terperinci tentang efektivitas alat dalam memastikan kepatuhan dan transparansi untuk berbagai model AI.
 8. **Studi Perbandingan Prinsip Etika AI di Berbagai Industri:** Proyek ini mengharuskan mahasiswa untuk melakukan studi perbandingan prinsip dan praktik etika AI di berbagai industri seperti perawatan kesehatan, keuangan, dan pemasaran. Mahasiswa akan menganalisis persyaratan etika dan praktik terbaik khusus industri, mensurvei praktisi AI, dan membuat laporan komprehensif tentang bagaimana prinsip-prinsip etika diterapkan secara berbeda di berbagai sektor ini. Laporan akhir harus mencakup rekomendasi untuk menyelaraskan praktik etika AI di berbagai industri.
 9. **Menjelajahi Peran Asal Usul Data dalam Sistem AI:** Tujuan proyek ini adalah untuk melacak dan mengelola asal usul data dalam sistem AI untuk memastikan integritas dan ketertelusuran data. Mahasiswa akan mengimplementasikan sistem silsilah data menggunakan teknologi seperti blockchain atau buku besar terdistribusi untuk melacak asal usul, transformasi, dan penggunaan data dalam model AI. Sebuah dasbor harus dikembangkan untuk memvisualisasikan perjalanan data, dan sebuah laporan



harus diberikan yang membahas manfaat dan keterbatasan penggunaan asal usul data dalam sistem AI.

10. **Merancang Sistem AI dengan Pengambilan Keputusan Etis Terintegrasi:** Dalam proyek ini, mahasiswa akan membangun sistem AI yang mengintegrasikan kemampuan pengambilan keputusan etis berdasarkan aturan moral dan teori etika yang telah ditentukan sebelumnya. Agen AI harus menggunakan pendekatan berbasis aturan atau pembelajaran mesin (ML) untuk membuat keputusan yang mempertimbangkan implikasi etis. Mahasiswa harus mengevaluasi kinerja sistem menggunakan berbagai dilema dan skenario etis, dan memberikan laporan terperinci tentang kemampuan sistem dalam menangani keputusan etis yang kompleks.



BAB 7

PENGANTAR PARADIGMA ALGORITMA CERDAS

7.1 PENGANTAR PARADIGMA CERDAS

Di era transformasi digital, algoritma cerdas menjadi penggerak inovasi dalam kesehatan, transportasi, pendidikan, dan infrastruktur perkotaan. Dengan dukungan kecerdasan buatan (AI), pembelajaran mesin (ML), dan Internet of Things (IoT), sistem dapat menganalisis data, menghasilkan keputusan, dan menjalankan tugas kompleks dengan intervensi manusia yang lebih sedikit. Bab ini membahas paradigma dasar untuk merancang dan menerapkan algoritma cerdas serta arah pengembangannya agar sistem dunia cerdas tetap terhubung, efisien, aman, dan dapat dipertanggungjawabkan.

Aplikasi dunia cerdas mencakup berbagai macam kasus penggunaan, seperti mesin pencari web canggih (misalnya, Google), sistem rekomendasi (digunakan oleh platform seperti YouTube, Amazon, dan Netflix), alat pengenalan suara (misalnya, Siri dan Alexa), kendaraan otonom, sistem pengambilan keputusan otomatis, dan agen AI yang mampu unggul dalam permainan strategis seperti catur dan Go. Dengan kemajuan berkelanjutan dalam perangkat keras komputasi, semakin banyak tugas kompleks yang kini dikategorikan sebagai membutuhkan kemampuan "cerdas". Bab ini mencakup area-area kunci berikut:

- **Paradigma Algoritma Standar:** Ini adalah model atau kerangka kerja generik yang membentuk dasar untuk merancang kelas algoritma. Paradigma ini mencakup metode bagi-dan-taklukkan, metode serakah, pemrograman dinamis, dan penelusuran balik.
- **Masalah Cerdas:** Pernyataan masalah yang terdefinisi dengan baik yang merangkum tantangan inti tanpa membahas penyebab mendasar atau menyarankan solusi langsung. Masalah-masalah ini sering muncul di lingkungan yang dinamis dan kaya data.
- **Solusi Cerdas:** Pendekatan holistik yang mengintegrasikan teknologi canggih, baik dalam perangkat keras (misalnya, perangkat IoT, sensor) maupun perangkat lunak (misalnya, model AI/ML), untuk mengatasi tantangan dunia nyata secara efisien dan adaptif.
- **Jenis Algoritma:** Ini mencakup algoritma eksak yang menjamin solusi optimal dan algoritma aproksimasi yang digunakan dalam skenario di mana solusi eksak secara komputasi tidak layak atau tidak ada.
- **Wawasan Berbasis Data:** Teknik dan metodologi yang digunakan untuk mengekstrak kesimpulan yang dapat ditindaklanjuti dari sistem yang berpusat pada data, memungkinkan peningkatan dan optimasi kinerja algoritma secara berkelanjutan.

Dengan memeriksa aspek-aspek ini, bab ini bertujuan untuk memberikan pemahaman komprehensif tentang desain, aplikasi, dan lintasan masa depan algoritma dunia cerdas dalam membangun ekosistem digital yang cerdas dan saling terhubung.



Paradigma algoritma cerdas berfungsi sebagai pola desain yang menghubungkan jenis masalah dengan strategi penyelesaiannya. Pemilihan paradigma harus mempertimbangkan struktur data, ketidakpastian, kebutuhan waktu nyata, keterbatasan energi, serta tingkat interpretabilitas yang dibutuhkan. Algoritma yang paling akurat belum tentu paling tepat apabila tidak dapat dijalankan pada perangkat sasaran atau tidak dapat diaudit. Konsep *dunia cerdas* mengacu pada ekosistem di mana sistem cerdas, yang didukung oleh sejumlah besar data, bekerja secara harmonis untuk memberikan solusi waktu nyata terhadap tantangan yang kompleks. Sistem ini dirancang untuk memahami lingkungannya, belajar darinya, dan menyesuaikan responsnya berdasarkan perubahan kontekstual.

Munculnya kota cerdas, sistem perawatan kesehatan cerdas, dan kendaraan otonom adalah contoh bagaimana teknologi ini membentuk kembali dunia. Inti dari transformasi ini adalah *algoritma dunia cerdas*, yang memanfaatkan teknik berbasis data untuk memberikan otomatisasi, efisiensi, dan kecerdasan dalam operasi. Paradigma dunia cerdas mencakup berbagai domain, mulai dari pengelolaan infrastruktur fisik hingga peningkatan proses pengambilan keputusan di lingkungan yang dinamis. Perkembangan ini terutama didorong oleh kemajuan dalam:

- AI dan ML
- IoT
- Analisis data dan big data
- Komputasi awan dan edge computing
- Sistem terdistribusi
- Seiring konvergensi teknologi-teknologi ini, kebutuhan akan algoritma yang efisien dan terukur menjadi semakin penting untuk memenuhi tuntutan lingkungan yang kompleks dan saling terhubung.

7.2 FONDASI ALGORITMA CERDAS

Setiap kelompok algoritma memiliki asumsi, kekuatan, dan keterbatasan. Pembelajaran mesin unggul pada pengenalan pola, optimasi mencari konfigurasi terbaik, sedangkan algoritma terdistribusi menjaga koordinasi antarperangkat. Sistem dunia cerdas sering memerlukan kombinasi ketiganya, sehingga integrasi antarmodul dan pengendalian kegagalan menjadi bagian penting dari desain. Algoritma yang mendorong paradigma cerdas bervariasi dalam kompleksitas dan aplikasinya, tetapi memiliki tujuan umum yaitu kecerdasan, otonomi, dan efisiensi. Beberapa algoritma kunci yang membentuk dasar sistem cerdas meliputi:

Algoritma Pembelajaran Mesin (ML)

- **Algoritma Pembelajaran Terawasi:** Algoritma ini belajar dari data berlabel, dan contohnya termasuk pohon keputusan, mesin vektor pendukung, dan jaringan saraf.
 - **Algoritma Pembelajaran Tak Terawasi:** Algoritma tak terawasi, seperti pengelompokan k-means dan analisis komponen utama, menganalisis data tanpa label yang telah ditentukan sebelumnya untuk mengungkap pola dan hubungan tersembunyi.
- Pembelajaran Penguatan (RL):** Algoritma RL sangat penting untuk mengembangkan agen otonom yang belajar membuat keputusan berdasarkan umpan balik dari



lingkungannya. Algoritma ini sangat penting dalam robotika, transportasi cerdas, dan aplikasi game.

Algoritma Optimasi

Teknik optimasi memainkan peran penting dalam meningkatkan kinerja sistem dunia cerdas. Beberapa contohnya meliputi:

- **Algoritma Genetika:** Algoritma evolusioner ini digunakan untuk menemukan solusi optimal atau mendekati optimal dengan meniru proses seleksi alam.
- **Optimasi Swarm Partikel:** Ini adalah teknik optimasi berbasis populasi yang terinspirasi oleh perilaku sosial kawanan burung atau gerombolan ikan.

IoT dan Algoritma Terdistribusi

Dengan semakin meningkatnya penyebaran perangkat IoT, algoritma yang menangani pemrosesan data terdistribusi dan pengambilan keputusan sangat penting:

- **Algoritma Konsensus Terdistribusi:** Algoritma seperti Paxos dan Raft memastikan konsistensi dan keandalan dalam sistem terdistribusi.
- **Algoritma Komputasi Tepi:** Algoritma ini berfokus pada pemrosesan data di tepi jaringan, mengurangi latensi dan mengoptimalkan kinerja aplikasi waktu nyata.

7.3 PETA JALAN UNTUK KEMAJUAN MASA DEPAN

Peta jalan pengembangan perlu menyeimbangkan peningkatan kinerja dengan keamanan, privasi, interoperabilitas, dan keberlanjutan. Skalabilitas tanpa tata kelola dapat memperluas dampak kesalahan, sedangkan otonomi tanpa mekanisme audit dapat melemahkan akuntabilitas. Karena itu, kemajuan teknis harus disertai standar pengujian dan pemantauan yang sebanding. Masa depan algoritma dunia cerdas dibentuk oleh evolusi berkelanjutan dari daya komputasi, ketersediaan data, dan meningkatnya permintaan akan otomatisasi. Area fokus utama meliputi hal-hal berikut:

Meningkatkan Skalabilitas

Seiring dengan meluasnya skala sistem cerdas, pengembangan algoritma yang dapat menangani jaringan luas perangkat yang saling terhubung dan aliran data secara efisien akan menjadi sangat penting. Kemajuan di masa mendatang akan berfokus pada peningkatan skalabilitas dan interoperabilitas sistem di berbagai industri.

Privasi dan Keamanan Data

Dengan menjamurnya perangkat IoT dan sistem berbasis data, masalah privasi dan keamanan menjadi hal yang terpenting. Algoritme masa depan harus menggabungkan enkripsi tingkat lanjut, manajemen data terdesentralisasi, dan teknik menjaga privasi seperti privasi diferensial.

Pengambilan Keputusan yang Otonom dan Cerdas

Pengembangan algoritma pengambilan keputusan yang lebih otonom dan cerdas akan menjadi kunci dalam aplikasi seperti kendaraan otonom, kota pintar, dan layanan kesehatan. Integrasi model RL mendalam dan AI tingkat lanjut akan sangat penting bagi evolusi sistem ini.



Komputasi Ramah Lingkungan dan Efisiensi Energi

Algoritme dunia pintar juga harus fokus pada efisiensi energi, memastikan bahwa sumber daya komputasi besar yang diperlukan untuk pemrosesan data dan otomatisasi tidak mengorbankan kelestarian lingkungan. Algoritma yang sadar energi dan paradigma komputasi ramah lingkungan akan menjadi semakin penting.

7.4 KESIMPULAN

Algoritme dunia pintar mewakili masa depan inovasi di berbagai sektor, mulai dari layanan kesehatan hingga manajemen perkotaan. Dengan memahami paradigma yang mendorong algoritma ini—berfokus pada bidang skalabilitas, keamanan, dan otonomi—kita dapat membuka jalan bagi masa depan sistem cerdas yang saling terhubung. Peta jalan ini menyoroti bidang-bidang utama yang memerlukan kemajuan untuk sepenuhnya mewujudkan potensi dunia yang cerdas dan berbasis data.



BAB 8

ALGORITMA OPTIMASI

8.1 OPTIMASI DENGAN KETERBATASAN

Optimasi merupakan proses mencari rancangan terbaik, baik dengan meminimalkan biaya produksi maupun memaksimalkan efisiensi. Algoritma optimasi membandingkan alternatif secara sistematis sampai ditemukan solusi terbaik atau solusi yang memenuhi kriteria tertentu. Bab ini membahas pemrograman dinamis, branch and bound, pemrograman linier, serta metode simpleks, elipsoid, dan titik interior, termasuk penerapannya pada persoalan nyata seperti alokasi sumber daya dan penjadwalan awak pesawat.

- **Memahami Teknik Optimasi:** Pelajari prinsip-prinsip inti di balik berbagai algoritma optimasi dan aplikasinya.
- **Memecahkan Masalah Dunia Nyata:** Terapkan algoritma ini untuk memecahkan masalah praktis, seperti alokasi sumber daya dan penjadwalan.
- **Mengembangkan Solusi yang Efisien:** Jelajahi metode untuk meningkatkan kinerja dan efisiensi proses optimasi.

Dalam optimasi dengan kendala, solusi terbaik tidak dicari pada seluruh ruang kemungkinan, tetapi hanya pada wilayah layak yang memenuhi seluruh pembatas. Kendala dapat merepresentasikan kapasitas fisik, anggaran, aturan keselamatan, atau persyaratan layanan. Perbedaan antara fungsi objektif dan wilayah layak penting karena solusi dengan nilai objektif paling baik tetap tidak berguna apabila melanggar kendala. Optimasi bertujuan menemukan nilai terbaik dari suatu fungsi, misalnya meminimalkan biaya atau memaksimalkan keuntungan. Dalam praktiknya, keputusan hampir selalu dibatasi oleh kapasitas, permintaan, anggaran, keselamatan, atau persyaratan teknis. Sebuah pabrik, misalnya, tidak dapat menghentikan seluruh produksi hanya untuk menekan biaya karena tetap harus memenuhi permintaan. Situasi semacam ini disebut optimasi dengan kendala.

Definisi 8.1 (*Optimasi Terbatas*)

Tujuannya adalah meminimalkan fungsi $f(x)$ dengan memenuhi kendala $h(x) = b$, untuk $x \in X$. Di sini, x merupakan variabel keputusan yang dapat disesuaikan, f adalah fungsi objektif yang hendak diminimalkan, h dan b merepresentasikan kendala, sedangkan X mendefinisikan himpunan nilai x yang diperbolehkan.

Pembahasan menggunakan vektor, tetapi penulisannya tidak selalu dicetak tebal demi kesederhanaan. Formulasi tersebut mencakup sebagian besar skenario optimasi. Masalah maksimasi dapat diubah menjadi minimasi dengan meminimalkan $-f(x)$. Kendala ketaksamaan, misalnya $h(x) \geq b$, dapat diubah menjadi kendala persamaan melalui variabel slack yang sesuai. Oleh sebab itu, formulasi ini cukup fleksibel untuk merepresentasikan beragam masalah. Pemrograman linier adalah kasus khusus dimana segala sesuatunya linier. Kita dapat menuliskannya sebagai: Minimalkan $c^T x$ dengan tunduk pada sejumlah



ketaksamaan dan persamaan linier serta kendala nonnegativitas pada x . Ini dapat disederhanakan menjadi dua bentuk standar:

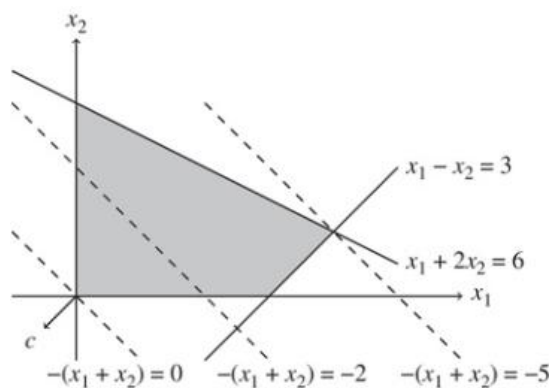
Definisi 8.2 (Bentuk Umum dan Standar dari Program Linear)

Bentuk umum adalah: minimalkan $c^T x$ dengan syarat $Ax \geq b$ dan $x \geq 0$. Bentuk standar adalah: minimalkan $c^T x$ dengan syarat $Ax = b$ dan $x \geq 0$.

Bentuk-bentuk ini setara dan dapat mewakili masalah pemrograman linier apa pun. Variabel yang tidak dibatasi dapat dinyatakan sebagai selisih dari dua variabel non-negatif. Bentuk-bentuk yang disederhanakan ini berguna untuk mengembangkan teori tentang program linier, meskipun tidak selalu menyederhanakan pemecahan masalah praktis.

Contoh 8.1

Mari kita minimalkan $-(x_1 + x_2)$ dengan kendala $x_1 + 2x_2 \leq 6$, $x_1 - x_2 \leq 3$, dan $x_1, x_2 \geq 0$. Kita dapat memvisualisasikan ini dalam 2D. Daerah yang layak adalah daerah di mana semua kendala terpenuhi. Nilai minimum ditemukan dengan bergerak sepanjang arah vektor biaya, yang terjadi pada titik persimpangan garis-garis kendala.



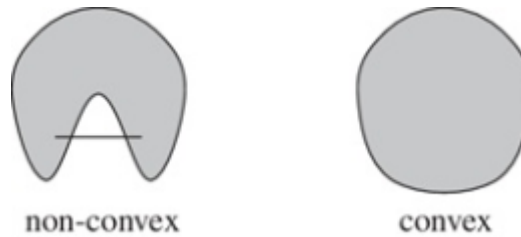
Menyelesaikan masalah umum yang dibatasi itu sulit. Namun, kita dapat menyelesaikan masalah optimasi yang tidak dibatasi.

8.2 METODE MINIMUM GLOBAL

Sifat cembung menyederhanakan optimasi karena setiap minimum lokal sekaligus merupakan minimum global. Konsekuensi ini memungkinkan penggunaan metode berbasis gradien dengan jaminan yang lebih kuat daripada pada fungsi non-cembung. Namun, jaminan tersebut tetap bergantung pada ketepatan asumsi mengenai fungsi objektif, daerah layak, dan sifat matriks Hessian. Untuk masalah tanpa kendala, syarat perlu agar x^* meminimalkan f adalah gradien fungsi pada titik tersebut bernilai nol, yaitu $\nabla f(x^*) = 0$. Namun, syarat ini belum cukup; diperlukan sifat cembung agar titik stasioner tersebut dapat dijamin sebagai minimum global.

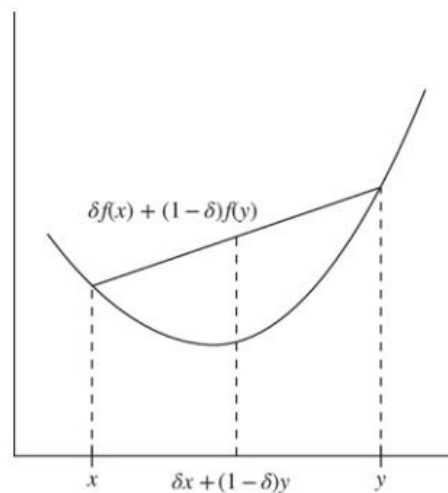
Definisi 8.3 (Daerah Cembung)

Suatu daerah dikatakan cembung jika, untuk setiap dua titik di dalamnya, segmen garis yang menghubungkan keduanya juga berada di daerah tersebut.



Definisi 8.4 (Fungsi Cembung)

Suatu fungsi dikatakan cembung jika grafiknya terletak di bawah setiap segmen garis yang menghubungkan dua titik pada grafik. Fungsi cekung adalah kebalikan dari fungsi cembung.



Untuk fungsi yang dapat didiferensialkan dua kali, kekonveksian dapat diperiksa menggunakan matriks Hessian.

Lemma 8.1

Suatu fungsi yang dapat didiferensialkan dua kali adalah cembung jika matriks Hessian-nya positif semidefinit.

Definisi 8.5 (Positif Semidefinit)

Suatu matriks H disebut positif semidefinit apabila $v^T H v \geq 0$ untuk setiap vektor v .

Teorema 8.1

Jika gradien bernilai nol dan matriks Hessian positif semidefinit pada x^* , maka x^* meminimalkan f pada suatu daerah cembung.

Fungsi linear bersifat cembung, yang bermanfaat. Namun, kita perlu mengubah masalah yang dibatasi menjadi masalah yang tidak dibatasi untuk menerapkan metode ini.

8.3 MENYELESAIKAN PERSAMAAN LINEAR

Sistem persamaan linier merupakan komponen dasar dalam simulasi, estimasi, kontrol, dan optimasi. Pada masalah berskala besar dan jarang, metode langsung dapat terlalu mahal sehingga metode iteratif lebih sesuai. Kinerja metode iteratif sangat ditentukan oleh bilangan kondisi dan kualitas prakondisioner yang digunakan. Eliminasi Gaussian menyelesaikan sistem



$Ax = b$ dalam waktu $O(n^3)$. Untuk matriks simetris positif semidefinit A , tersedia metode iteratif. Setiap iterasi melibatkan perkalian matriks-vektor, sedangkan laju konvergensi bergantung pada bilangan kondisi $\kappa = \lambda_{\max}/\lambda_{\min}$. Kompleksitas iterasi penurunan tercuram adalah $O(\kappa \log(1/\epsilon))$, sedangkan gradien konjugasi membutuhkan $O(\sqrt{\kappa} \log(1/\epsilon))$.

Prakondisi bertujuan mempercepat metode iteratif dengan menyelesaikan sistem $B^{-1}Ax = B^{-1}b$, di mana B^{-1} dipilih agar sistem hasil transformasi lebih mudah diselesaikan. Secara ideal, B menyerupai A , tetapi tetap mudah diinverskan secara komputasional. Kita akan meninjau metode iteratif, kemudian menggunakan pohon rentang peregang rendah sebagai pra-kondisioner untuk sistem Laplacian, dan akhirnya mengeksplorasi bagaimana penambahan tepi ke pra-kondisioner pohon mengarah ke solusi waktu mendekati linear.

Penurunan Tercuram: Minimisasi Berbasis Gradien

Penurunan tercuram mencari minimum fungsi dengan bergerak mengikuti arah turunan negatif. Penyelesaian $Ax = b$ setara dengan meminimalkan fungsi kuadratik $f(x) = 1/2 x^T Ax - b^T x + c$. Sifat positif semidefinit menjamin bahwa titik yang diperoleh merupakan minimum, bukan titik pelana. Dimulai dari x_0 , setiap iterasi memperbarui solusi melalui $x_{i+1} = x_i + \alpha_i r_i$, dengan $r_i = b - Ax_i$ sebagai arah gradien negatif. Nilai α_i dipilih agar $r_{i+1}^T r_i = 0$. Kondisi ortogonalitas ini memastikan pergerakan dilanjutkan hingga gradien baru tegak lurus terhadap arah saat ini. Nilai optimal dihitung sebagai:

$$r_{i+1} = b - Ax_{i+1} = r_i - \alpha_i Ar_i,$$

$$r_i^T r_{i+1} = 0 \Rightarrow \alpha_i = \frac{r_i^T r_i}{r_i^T Ar_i}.$$

Metode penurunan tercuram banyak digunakan dan mudah dihitung untuk sistem linier. Setiap iterasi membutuhkan waktu $O(m)$, dengan m menyatakan jumlah entri tak nol dalam A . Persoalan utama kemudian terletak pada jumlah iterasi yang diperlukan sampai galat berada di bawah toleransi. Dalam contoh dua dimensi berbentuk elips, konvergensi memerlukan kira-kira λ_2/λ_1 langkah, dengan λ_1 dan λ_2 menyatakan sumbu elips. Secara umum, penurunan tercuram konvergen dalam $O(\kappa)$ iterasi. Jika $e_i = x_i - x^*$ menyatakan galat pada iterasi ke- i , maka galat tersebut memenuhi hubungan berikut:

$$e_{i+1} = x_{i+1} - x^* = (I - \alpha_i A)e_i.$$

Menganalisis konvergensi untuk suatu nilai tetap α sudah cukup.

Dengan melakukan ekspansi e_0 pada basis spektral, kita peroleh

$$e_i = (I - \alpha A)^i e_0 = \sum_{j \in [n]} (1 - \alpha \lambda_j)^i \beta_j v_j$$

Konvergensi ditentukan oleh $\max(|1 - \alpha \lambda_{\min}|, |1 - \alpha \lambda_{\max}|)$. Pemilihan $\alpha = \frac{2}{\lambda_{\min} + \lambda_{\max}}$ menghasilkan batas galat:



$$\|e_i\|^2 \leq \left(1 - \frac{2\lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right)^{2i} \|e_0\|^2 \leq e^{-\frac{4i\lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}} \|e_0\|^2$$

Setelah $O(\kappa)$ iterasi, galat berkurang dengan faktor konstan. Dengan demikian, untuk mencapai toleransi ϵ diperlukan $O(\kappa \log(1/\epsilon))$ iterasi. Metode gradien konjugasi konvergen lebih cepat, yaitu dalam $O(\sqrt{\kappa} \log(1/\epsilon))$ iterasi, meskipun pembuktiannya tidak dibahas pada bagian ini.

Meningkatkan Konvergensi

Prakondisi melibatkan pencarian matriks B yang mudah diinverskan dan secara spektral menyerupai A . *Mudah Dibalik*: Matriks kedekatan pohon berbobot mudah dibalik melalui substitusi balik. Daun pohon sesuai dengan satu baris entri bukan nol, membuat persamaan dapat diselesaikan secara langsung. Pohon dapat diuraikan dengan menghilangkan daun, memungkinkan substitusi berurutan. Invers tidak perlu dihitung secara eksplisit. Istilah “*mudah diinverskan*” berarti tersedia prosedur waktu linier untuk memperoleh $A_T^{-1}z$ dari suatu z . Dengan demikian, penurunan gradien terprakondisi dapat menyelesaikan $A_T^{-1}Ax = A_T^{-1}b$ secara efisien. Prinsip yang sama berlaku pada matriks Laplacian pohon. Matriks B dikatakan “*menyerupai*” A apabila $B^{-1}A$ memiliki bilangan kondisi yang lebih kecil. Matriks positif semidefinit berhubungan dengan bentuk kuadrat: apabila $x^T Bx \geq x^T Ax$ untuk setiap x , ditulis $B \succcurlyeq A$.

Teorema 8.2

Jika $kB \succcurlyeq A \succcurlyeq B \succcurlyeq 0$, maka seluruh nilai eigen dari $B^{-1}A$ berada dalam interval $1, k$.

Bukti:

Pembuktian menggunakan dua fakta aljabar linier: (1) matriks positif semidefinit memiliki akar kuadrat; dan (2) matriks AB dan BA mempunyai himpunan nilai eigen yang sama. Cukup dibuktikan pernyataan tersebut untuk $B^{-1/2}AB^{-1/2}$. Dengan menggunakan hasil bagi Rayleigh, diperoleh:

$$\frac{x^T B^{-1/2}AB^{-1/2}x}{x^T x} = \frac{y^T Ay}{y^T By} \in [1, k],$$

dengan $y = B^{-1/2}x$. Dengan demikian, nilai eigen $B^{-1}A$ berada pada interval $1, k$.

Prakondisi dengan Pohon

Pohon rentang T merupakan prakondisioner yang baik apabila $kL_T \succcurlyeq L_G \succcurlyeq L_T$. Karena T adalah subgraf dari G , berlaku $L_G \succcurlyeq L_T$. Oleh karena itu, bagian yang perlu dibuktikan adalah $kL_T \succcurlyeq L_G$. Hal ini dibuktikan menggunakan ketidaksamaan Cauchy–Schwarz:

$$(x_1 - x_{k+1})^2 \leq k((x_1 - x_2)^2 + \dots + (x_k - x_{k+1})^2).$$

Untuk setiap sisi $(u, v) \in G$, berlaku:



$$L_{(u,v)} \leq d_T(u, v)L_T.$$

Dengan menjumlahkan seluruh sisi, diperoleh $L_G \leq \text{st}(T)L_T$, dengan $\text{st}(T) = \sum_{(u,v) \in E} d_T(u, v)$. Untuk pohon berbobot, $d_T(u, v)$ diganti dengan $\sum 1/w_i$, dengan w_i sebagai bobot sisi pada lintasan yang bersangkutan.

Pohon rentang berbobot dengan peregangan $\tilde{O}(m \log n)$ dapat ditemukan dalam waktu mendekati linier. Gradien konjugasi dengan prakondisioner ini konvergen dalam $\tilde{O}(\sqrt{m})$ iterasi, dan setiap iterasi membutuhkan waktu $O(m)$. Dengan demikian, pemecah berkompleksitas $\tilde{O}(m^{3/2})$ ini lebih baik daripada eliminasi Gaussian pada graf jarang.

8.4 DUALITAS PEMROGRAMAN LINIER

Dualitas menghubungkan suatu program linier dengan masalah pasangan yang menyediakan batas terhadap nilai objektif. Solusi dual dapat berfungsi sebagai sertifikat optimalitas dan, dalam interpretasi ekonomi, sebagai harga bayangan bagi sumber daya atau kendala. Hubungan primal-dual juga menjadi dasar banyak algoritma aproksimasi dan optimasi daring. Dualitas pemrograman linier membentuk dasar bagi banyak topik yang telah kita bahas di kelas. Dalam kuliah ini, kita secara formal memperkenalkan konsep dualitas dan menghubungkannya dengan diskusi sebelumnya, termasuk kemacetan tol dan masalah pencocokan bobot maksimum. Kita mulai dengan menyajikan sepasang program linier primal dan dual dalam bentuk standar:

LP Primal	LP Dual
minimalkan $c^T x$	maksimalkan $y^T b$
dengan syarat $Ax \geq b$	dengan syarat $y^T A \leq c^T$
$x \geq 0$	$y \geq 0$

Misalkan nilai objektif optimal primal dilambangkan dengan α . Untuk menetapkan batas atas $\alpha \leq \delta$, cukup ditemukan solusi layak x yang memenuhi $c^T x = \delta$. Namun, motivasi utama dualitas adalah arah sebaliknya, yaitu membuktikan batas bawah $\alpha \geq \delta$. Untuk mengeksplorasi hal ini, pertimbangkan perluasan kendala kompak menjadi ketidaksetaraan atas variabel:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\geq b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\geq b_2, \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\geq b_m. \end{aligned}$$



Misalkan setiap kendala dikalikan dengan skalar positif $y_i > 0$ untuk $i \in [m]$. Karena skalar tersebut positif, arah ketaksamaan tetap dipertahankan. Dengan menjumlahkan seluruh kendala, untuk setiap x yang layak diperoleh:

$$\sum_{i=1}^m y_i \left(\sum_{j=1}^n a_{ij} x_j \right) \geq \sum_{i=1}^m y_i b_i = y^T b.$$

Menulis ulang dengan mengubah urutan penjumlahan:

$$\sum_{j=1}^n \left(\sum_{i=1}^m a_{ij} y_i \right) x_j \geq y^T b.$$

Apabila $\sum_{i=1}^m a_{ij} y_i \leq c_j$ untuk setiap j , maka $c^T x \geq y^T b$ untuk seluruh x yang layak. Dengan demikian, $y^T b$ menjadi batas bawah bagi nilai objektif primal.

Dengan demikian, program linear dual dapat dipahami sebagai masalah memaksimalkan batas bawah ini. Menurut teorema dualitas kuat, nilai optimal dari program primal dan dual adalah sama, asalkan keduanya layak dan terbatas. Oleh karena itu, solusi layak untuk dual memberikan batas bawah, sedangkan solusi layak untuk primal memberikan batas atas. Sebagai kasus khusus, pertimbangkan program linier primal tak terbatas: $\min -x_1$, dengan kendala $x_1 - x_2 \leq 100$ serta $x_1, x_2 \geq 0$. Program dual yang bersesuaian dapat diverifikasi tidak layak.

Kelonggaran Komplementer

Kelonggaran komplementer menawarkan metode untuk memverifikasi optimalitas pasangan solusi primal-dual x, y . Jika x, y optimal untuk pasangan program linier primal dan dual, maka:

$$\begin{aligned} y_i (Ax - b)_i &= 0 \quad \forall i \in [m], \\ x_j (c_j - (A^T y)_j) &= 0 \quad \forall j \in [n]. \end{aligned}$$

Bukti:

Karena x dan y layak untuk programnya masing-masing, berlaku:

$$c^T x \geq y^T Ax \geq y^T b.$$

Kesetaraan berlaku jika dan hanya jika pasangan x, y optimal. Dari $y^T Ax = y^T b$ diperoleh:

$$\sum_{i,j} y_i a_{ij} x_j = \sum_i y_i b_i \Rightarrow \sum_i y_i (Ax - b)_i = 0.$$

Karena setiap suku bernilai non-negatif, jumlahnya hanya akan nol jika setiap suku bernilai nol. Penurunan yang serupa menghasilkan kondisi kedua.



Minimisasi Kemacetan

Masalah minimisasi kemacetan dapat dirumuskan sebagai program linier dengan variabel aliran f_P pada setiap jalur P . Kendalanya adalah:

$$\sum_{P:e \in P} f_P \leq \mu,$$

$$\sum_{P \in \mathcal{P}(s_i, t_i)} f_P = 1$$

Masalah dualnya berkaitan dengan penetapan tarif tol. Terdapat variabel dual w_e untuk setiap sisi e dan variabel tak terbatas c_i untuk setiap pasangan (s_i, t_i) . Fungsi objektif dualnya adalah:

$$\max \sum_{i \in [k]} c_i - \mu \sum_e w_e.$$

Tunduk pada:

$$\forall P \in \mathcal{P}(s_i, t_i): \quad c_i - \sum_{e \in P} w_e \leq 0.$$

Kendala tersebut menjamin bahwa c_i tidak melebihi panjang jalur mana pun dari s_i ke t_i menurut tol $w_e \geq 0$. Dengan normalisasi $\sum_e w_e = 1$, bentuk dual menjadi:

$$\max \sum_i w(p_i)$$

$$\text{subject to } w_e \geq 0, \quad \sum_e w_e = 1$$

Formulasi ini memaksimalkan panjang rata-rata jalur terpendek di bawah metrik tol, secara efektif memodelkan kemacetan. Dari kelonggaran komplementer: (1) Hanya sisi dengan kemacetan maksimal yang memiliki tol bukan nol; dan (2) Aliran hanya terjadi di sepanjang jalur terpendek di bawah metrik tol. Prinsip ini tercermin dalam strategi pembelajaran online seperti algoritma pakar, di mana dinamika primal-dual dieksploitasi untuk menurunkan algoritma yang efisien.



Pencocokan Bobot Maksimum

Latihan: Turunkan formulasi primal dan dual untuk masalah pencocokan bobot maksimum yang dibahas dalam Kuliah 3. Analisis apa yang diungkapkan kelonggaran komplementer tentang struktur pencocokan optimal.

Permainan dan Solusi Strategis

Definisi 8.6 (*Permainan Bimatrix*)

Permainan dua pemain direpresentasikan oleh dua matriks $P, Q \in \mathbb{R}^{m \times n}$. Pemain baris memilih baris i , sedangkan pemain kolom memilih kolom j , tanpa mengetahui pilihan lawannya. Imbalan masing-masing pemain adalah P_{ij} dan Q_{ij} .

Contoh 8.2 (*Batu-kertas-gunting*)

Matriks pembayaran:

$$P = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}.$$

Hal ini biasanya disajikan dalam tabel pembayaran:

	R	P	S
R	(0,0)	(-1,1)	(1,-1)
P	(1,-1)	(0,0)	(-1,1)
S	(-1,1)	(1,-1)	(0,0)

Definisi 8.7 (*Strategi*)

Rangkaian strateginya adalah:

$$X = \left\{ x \in \mathbb{R}^m : x \geq 0, \sum_i x_i = 1 \right\}, \quad Y = \left\{ y \in \mathbb{R}^n : y \geq 0, \sum_j y_j = 1 \right\}.$$

Imbalan yang diharapkan bagi pemain baris adalah $p(x, y) = x^T P y$.

Definisi 8.8 (*Keseimbangan*)

Pasangan strategi $(x, y) \in X \times Y$ disebut keseimbangan apabila setiap strategi merupakan respons terbaik terhadap strategi lawannya:

$$p(x, y) \geq p(x', y) \quad \forall x' \in X; \quad q(x, y) \geq q(x, y') \quad \forall y' \in Y.$$

Teorema 8.3

Setiap permainan bimatriks hingga memiliki setidaknya satu profil strategi keseimbangan.

Teorema Minimax

Salah satu kelas permainan yang terkenal dalam teori permainan adalah *permainan zero-sum*.



Definisi 8.9 (Permainan Zero-sum)

Suatu permainan bimatriks disebut *permainan jumlah-nol* atau *permainan matriks* apabila untuk setiap i, j berlaku $q_{ij} = -p_{ij}$. Dalam keadaan tersebut, jumlah imbalan kedua pemain selalu nol.

Dalam permainan matriks, cukup dengan menentukan satu matriks pembayaran saja, karena matriks lawan hanyalah invers aditif dari matriks tersebut.

Contoh 8.3

Permainan klasik batu-kertas-gunting, seperti yang telah diperkenalkan sebelumnya, adalah contoh dari permainan zero-sum.

Teorema 8.4

Misalkan $P \in \mathbb{R}^{m \times n}$ menyatakan matriks pembayaran. Maka,

$$\max_{x \in X} \min_{y \in Y} p(x, y) = \min_{y \in Y} \max_{x \in X} p(x, y).$$

Hal ini dapat dituliskan secara setara sebagai:

$$\max_{x \in X} \min_{y \in Y} p(x, y) = - \max_{y \in Y} \min_{x \in X} [-p(x, y)].$$

Sisi kiri mewakili imbalan terburuk terbaik yang dapat diperoleh pemain baris menggunakan strategi minimax. Sisi kanan melakukan hal yang sama untuk pemain kolom. Teorema ini membuktikan bahwa ketika kedua pemain mengadopsi strategi minimax masing-masing, permainan mencapai keseimbangan.

Bukti:

Nilai optimal dapat diperoleh dengan menyelesaikan program linier berikut:

Maksimumkan v dengan batasan:

$$\begin{aligned} \sum_{i=1}^m x_i p_{ij} &\geq v \quad \text{for all } j = 1, \dots, n, \\ \sum_{i=1}^m x_i &= 1, \\ x &\geq 0. \end{aligned}$$

Dengan memperkenalkan variabel slack $z \in \mathbb{R}^n$ dan $z \geq 0$, fungsi Lagrange dapat dituliskan sebagai:

$$L(v, x, z, w, y) = v + \sum_{j=1}^n y_j \left(\sum_{i=1}^m x_i p_{ij} - z_j - v \right) - w \left(\sum_{i=1}^m x_i - 1 \right).$$

Di sini, $w \in \mathbb{R}$ dan $y \in \mathbb{R}^n$ merupakan pengali Lagrange. Setelah disederhanakan, diperoleh:



$$\left(1 - \sum_{j=1}^n y_j\right) v + \sum_{i=1}^m \left(\sum_{j=1}^n p_{ij} y_j - w\right) x_i - \sum_{j=1}^n y_j z_j + w.$$

Agar fungsi Lagrange mempunyai batas bawah untuk semua $v \in R$ dan $x \geq 0$, kendala berikut harus dipenuhi:

$$\sum_{j=1}^n y_j = 1, \quad \sum_{j=1}^n p_{ij} y_j \leq w \quad \forall i, \quad y \geq 0.$$

Program linear ganda menjadi:

Minimalkan w dengan batasan:

$$\begin{aligned} \sum_{j=1}^n p_{ij} y_j &\leq w \quad \text{for all } i, \\ \sum_{j=1}^n y_j &= 1, \\ y &\geq 0. \end{aligned}$$

Formulasi ini sesuai dengan pemain kolom yang memilih strategi campuran untuk memastikan bahwa imbalan yang diharapkan tidak melebihi w . Solusi optimal masalah dual tersebut sama dengan $\min_{y \in Y} \max_{x \in X} p(x, y)$. Oleh karena dualitas kuat, Teorema Minimax terbukti.

Definisi 8.10 (Nilai)

Nilai dari permainan matriks dengan matriks pembayaran didefinisikan sebagai:

$$v = \max_{x \in X} \min_{y \in Y} p(x, y) = \min_{y \in Y} \max_{x \in X} p(x, y).$$

Secara umum, strategi keseimbangan dari permainan semacam itu memenuhi kondisi berikut:

Teorema 8.5

Pasangan $(x, y) \in X \times Y$ membentuk keseimbangan permainan matriks dengan matriks pembayaran P jika dan hanya jika:

$$\begin{aligned} \min_{y' \in Y} p(x, y') &= \max_{x' \in X} \min_{y' \in Y} p(x', y'), \\ \max_{x' \in X} p(x', y) &= \min_{y' \in Y} \max_{x' \in X} p(x', y'). \end{aligned}$$

Dengan kata lain, x dan y masing-masing merupakan strategi optimal untuk formulasi $\max \min$ dan $\min \max$.



8.5 MASALAH JARINGAN

Graf menyediakan representasi ringkas untuk sistem yang tersusun atas entitas dan hubungan, seperti jaringan transportasi, komunikasi, energi, dan distribusi. Setelah sistem dinyatakan sebagai simpul dan sisi, berbagai persoalan dapat dianalisis melalui aliran, jalur, konektivitas, pohon rentang, dan biaya. Ketepatan definisi graf sangat penting karena perubahan arah sisi atau batas kapasitas dapat mengubah sifat solusi secara mendasar.

Definisi Kunci

Pada bagian ini, kita akan mengeksplorasi berbagai masalah yang dimodelkan menggunakan graf. Untuk melanjutkan, kita akan memperkenalkan beberapa konsep fundamental dari teori graf.

Definisi 8.11 (*Grafik/Jaringan Berarah*)

Graf atau jaringan berarah didefinisikan sebagai pasangan $G = (V, E)$, dengan V sebagai himpunan simpul dan $E \subseteq V \times V$ sebagai himpunan sisi berarah. Sisi $(u, v) \in E$ menyatakan hubungan terarah dari simpul u menuju simpul v .

Definisi 8.12 (*Derajat*)

Derajat simpul $u \in V$ adalah banyaknya simpul $v \in V$ yang memenuhi $(u, v) \in E$ atau $(v, u) \in E$.

Definisi 8.13 (*Jalan Setapak*)

Jalan setapak dari simpul u ke simpul v adalah urutan simpul $u = v_1, \dots, v_k = v$ sehingga setiap pasangan berurutan memenuhi $(v_i, v_{i+1}) \in E$. Pada *jalan tak berarah*, sisi boleh dilalui dalam kedua arah, yaitu $(v_i, v_{i+1}) \in E$ atau $(v_{i+1}, v_i) \in E$.

Memungkinkan pergerakan ke kedua arah.

Definisi 8.14 (*Jalur*)

Jalur adalah jalan setapak yang seluruh simpulnya $v_1, \dots, v_k$ berbeda.

Definisi 8.15 (*Siklus*)

Siklus adalah jalan setapak dengan simpul $v_1, \dots, v_{k-1}$ berbeda dan $v_1 = v_k$.

Definisi 8.16 (*Grafik Terhubung*)

Suatu grafik dikatakan *terhubung* jika terdapat jalur tak berarah antara setiap pasangan simpul.

Definisi 8.17 (*Pohon*)

Pohon adalah grafik terhubung yang tidak mengandung siklus tak berarah.

Definisi 8.18 (*Pohon Rentang*)

Pohon rentang dari graf $G = (V, E)$ adalah subgraf $G' = (V', E')$ yang memenuhi $V' = V$, $E' \subseteq E$, dan subgraf tersebut membentuk pohon.

Masalah Aliran Biaya Minimum

Pertimbangkan graf berarah $G = (V, E)$ dengan $|V| = n$ simpul. Misalkan $b \in \mathbb{R}^n$ menyatakan vektor aliran bersih. Setiap sisi $(i, j) \in E$ dikaitkan dengan biaya c_{ij} , batas bawah $\underline{m}_{ij}$, dan batas atas $\overline{m}_{ij}$ pada aliran yang diizinkan. Nilai-nilai ini direpresentasikan oleh matriks C , M , dan $\overline{M} \in \mathbb{R}^{n \times n}$. Setiap komponen b_i menyatakan aliran masuk atau keluar bersih pada simpul i . Nilai $b_i > 0$ menunjukkan bahwa simpul tersebut merupakan sumber, sedangkan $b_i < 0$ menunjukkan tujuan atau konsumen. Besaran b_i hanya menangkap aliran bersih yang



dihasilkan atau diserap pada simpul, bukan aliran transit. Matriks aliran dikatakan sebagai aliran biaya minimum jika meminimalkan total biaya transportasi, dengan tunduk pada kendala berikut:

Minimalkan $\sum_{(i,j) \in E} c_{ij} x_{ij}$ dengan tunduk pada:

$$b_i + \sum_{j:(j,i) \in E} x_{ji} = \sum_{j:(i,j) \in E} x_{ij} \quad \text{for all } i \in V,$$

$$\underline{m}_{ij} \leq x_{ij} \leq \overline{m}_{ij} \quad \text{for all } (i,j) \in E.$$

Masalah ini dapat diekspresikan sebagai program linier. Meskipun dapat ditulis dalam bentuk standar menggunakan matriks kendala yang besar, formulasi yang dihasilkan tidak efisien secara komputasi untuk jaringan skala besar. Oleh karena itu, algoritma yang lebih khusus lebih disukai. Syarat kelayakan yang diperlukan adalah:

$$\sum_{i \in V} b_i = 0,$$

memastikan bahwa total pasokan sama dengan total permintaan. Untuk menyederhanakan lebih lanjut, masalah dapat dirumuskan sebagai masalah sirkulasi dengan $b_i = 0$ untuk seluruh $i \in V$. Hal ini dilakukan dengan menambahkan simpul bantu tempat kelebihan pasokan atau permintaan dialihkan. Sebagai contoh, jika $b_i = -1$ menunjukkan permintaan bersih, satu unit aliran keluar dari simpul i diarahkan ke simpul bantu agar aliran seimbang.

Masalah aliran tanpa kapasitas memenuhi $\underline{m}_{ij} = 0$ dan $\overline{m}_{ij} = \infty$ untuk seluruh $(i,j) \in E$. Masalah tersebut dapat bersifat tak terbatas atau terbatas. Apabila terbatas, masalah dapat dimodelkan secara ekuivalen dengan menetapkan kapasitas hingga yang lebih besar daripada aliran maksimum yang diperlukan. Pada bagian selanjutnya, kita akan menunjukkan bagaimana masalah aliran biaya minimum ini dapat direduksi menjadi bentuk yang lebih sederhana dan sudah dikenal.

Masalah Transportasi

Masalah transportasi merupakan contoh khusus dari masalah aliran biaya minimum yang dirumuskan pada graf bipartit. Dalam struktur bipartit, himpunan simpul dipartisi menjadi dua himpunan terpisah, A dan B , sehingga seluruh sisi diarahkan dari simpul dalam A menuju simpul dalam B . Simpul pada A disebut pemasok, sedangkan simpul pada B disebut konsumen. Dalam pengaturan ini, masalah dapat dirumuskan sebagai:

Minimalkan $\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$ dengan kendala:

$$\sum_{j=1}^m x_{ij} = s_i \quad \text{for } i = 1, \dots, n,$$

$$\sum_{i=1}^n x_{ij} = d_j \quad \text{for } j = 1, \dots, m,$$



$$x_{ij} \geq 0.$$

Di sini, s_i menyatakan kapasitas pasokan pemasok i , sedangkan d_j menyatakan kebutuhan konsumen j . Vektor pasokan $s \in \mathbb{R}^n$ dan permintaan $d \in \mathbb{R}^m$ diasumsikan nonnegatif dan seimbang, yaitu:

$$s_i, d_j \geq 0, \quad \sum_{i=1}^n s_i = \sum_{j=1}^m d_j.$$

Matriks biaya $c \in \mathbb{R}^{n \times m}$ mendefinisikan biaya transportasi per unit dari pemasok i kepada konsumen j . Berikut ini, kami menunjukkan bahwa setiap masalah aliran biaya minimum terbatas dapat diubah menjadi masalah transportasi yang setara.

Teorema 8.6

Setiap masalah aliran biaya minimum dengan kapasitas hingga atau biaya nonnegatif dapat ditransformasikan menjadi masalah transportasi yang ekuivalen.

Bukti:

Pertimbangkan masalah aliran biaya minimum pada jaringan berarah $G = (V, E)$. Tanpa mengurangi keumuman, batas bawah aliran diasumsikan nol, yaitu $\underline{m}_{ij} = 0$ untuk setiap $(i, j) \in E$. Apabila asumsi ini belum terpenuhi, lakukan transformasi berikut:

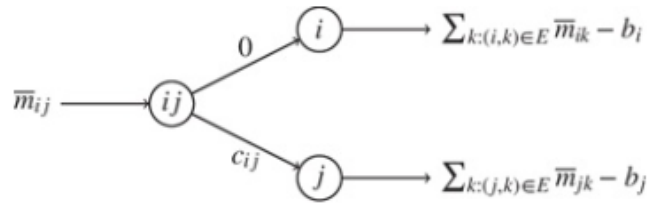
- $\underline{m}_{ij} := 0$.
- $\bar{m}_{ij} := \bar{m}_{ij} - \underline{m}_{ij}$.
- Sesuaikan keseimbangan simpul: $b_i := b_i - \underline{m}_{ij}$ dan $b_j := b_j + \underline{m}_{ij}$.
- Ubah variabel aliran menjadi $x_{ij} := x_{ij} - \underline{m}_{ij}$.

Secara intuitif, transformasi ini menetapkan aliran minimum pada setiap sisi, menyederhanakan masalah tanpa mengubah strukturnya. Lebih lanjut, seluruh kapasitas dapat diasumsikan hingga. Apabila suatu sisi memiliki kapasitas tak terbatas tetapi biaya nonnegatif, batas atas buatan seperti $\sum_{i \in V} |b_i|$ dapat diberlakukan tanpa mengubah optimalitas. Karena biayanya nonnegatif, solusi optimal menghindari siklus aliran berlebih dan menggunakan paling banyak $\sum_{i \in V} |b_i|$ unit secara keseluruhan. Sekarang kita membangun masalah transportasi yang setara:

- Untuk setiap simpul $i \in V$, perkenalkan konsumen dengan permintaan:

$$d_i := \sum_{k: (i,k) \in E} \bar{m}_{ik} - b_i$$

- Untuk setiap sisi $(i, j) \in E$, buat pemasok dengan pasokan $\bar{m}_{ij}$. Pemasok tersebut dihubungkan kepada:
 - Konsumen i dengan biaya transportasi 0;
 - Konsumen j dengan biaya transportasi c_{ij} .



Intuisi transformasi ini dapat dilihat pada sisi i, j . Apabila kapasitas sisi adalah 5 dan hendak dikirim 3 unit melaluinya, model transportasi mengirimkan 3 unit dari pemasok ij kepada konsumen j , sedangkan 2 unit yang tersisa dikirim kepada konsumen i .

Untuk memverifikasi bahwa kendala konservasi aliran tetap terjaga, perhatikan bahwa untuk setiap node, total aliran masuk dalam jaringan transportasi adalah:

$$\sum_{k:(i,k) \in E} (\bar{m}_{ik} - x_{ik}) + \sum_{k:(k,i) \in E} x_{ki}$$

Ekspresi ini sama dengan permintaan konsumen di simpul dalam jaringan baru jika dan hanya jika:

$$\sum_{k:(i,k) \in E} (\bar{m}_{ik} - x_{ik}) + \sum_{k:(k,i) \in E} x_{ki} = \sum_{k:(i,k) \in E} \bar{m}_{ik} - b_i$$

yang disederhanakan menjadi:

$$b_i + \sum_{k:(k,i) \in E} x_{ki} - \sum_{k:(i,k) \in E} x_{ik} = 0$$

Persamaan ini tepat merupakan kendala konservasi aliran pada simpul i dalam formulasi aliran biaya minimum semula. Oleh karena itu, transformasi tersebut sah dan lengkap.

Ini persisnya merupakan kendala konservasi aliran untuk simpul i dalam formulasi aliran biaya minimum asli. Oleh karena itu, transformasi ini valid dan lengkap. Untuk menganalisis masalah transportasi secara efektif, akan bermanfaat untuk memperkenalkan dua set pengali Lagrange: satu set λ_i yang sesuai dengan kendala pasokan (untuk pemasok), dan set lain μ_j untuk kendala permintaan (untuk konsumen). Fungsi Lagrangian untuk masalah transportasi kemudian diberikan oleh: Agar mudah dibayangkan, pengali Lagrange dapat dianggap sebagai “harga bayangan” dari pasokan dan permintaan. Jika harga bayangan ini sudah cocok dengan biaya pengiriman, perpindahan alokasi tidak lagi memberi penghematan.

$$L(x, \lambda, \mu) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m \lambda_i \left(s_i - \sum_{j=1}^n x_{ij} \right) - \sum_{j=1}^n \mu_j \left(d_j - \sum_{i=1}^m x_{ij} \right)$$



di mana x_{ij} menunjukkan jumlah yang diangkut dari pemasok i ke konsumen j , c_{ij} adalah biaya satuan transportasi, s_i adalah pasokan di sumber i , dan d_j adalah permintaan di tujuan j .

Pemilihan tanda yang berlawanan untuk kedua set pengali memastikan kejelasan dalam kondisi optimalitas. Penyederhanaan ekspresi menghasilkan:

$$L(x, \lambda, \mu) = \sum_{i=1}^n \sum_{j=1}^m (c_{ij} - \lambda_i + \mu_j) x_{ij} + \sum_{i=1}^n \lambda_i s_i - \sum_{j=1}^m \mu_j d_j$$

Dengan mempertimbangkan kendala nonnegatif $x_{ij} \geq 0$, Lagrangian memiliki minimum terbatas jika dan hanya jika:

$$c_{ij} - \lambda_i + \mu_j \geq 0 \quad \forall i, j$$

Kondisi ini dikenal sebagai kondisi kelayakan ganda.

Pada kondisi optimal, prinsip kelonggaran komplementer mensyaratkan:

$$(c_{ij} - \lambda_i + \mu_j) x_{ij} = 0 \quad \forall i, j$$

Untuk mempermudah perhitungan, kita menyusun variabel dan parameter dalam format tabel transportasi. Setiap baris mewakili pemasok dan setiap kolom mewakili konsumen:

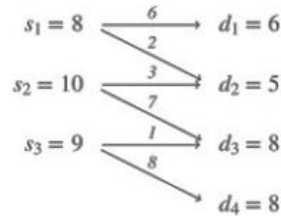
	μ_1	μ_2	μ_3	μ_4	
	$\lambda_1 - \mu_1$	$\lambda_1 - \mu_2$	$\lambda_1 - \mu_3$	$\lambda_1 - \mu_4$	
λ_1	x_{11} c_{11}	x_{12} c_{12}	x_{13} c_{13}	x_{14} c_{14}	s_1
	$\lambda_2 - \mu_1$	$\lambda_2 - \mu_2$	$\lambda_2 - \mu_3$	$\lambda_2 - \mu_4$	
λ_2	x_{21} c_{21}	x_{22} c_{22}	x_{23} c_{23}	x_{24} c_{24}	s_2
	$\lambda_3 - \mu_1$	$\lambda_3 - \mu_2$	$\lambda_3 - \mu_3$	$\lambda_3 - \mu_4$	
λ_3	x_{31} c_{31}	x_{32} c_{32}	x_{33} c_{33}	x_{34} c_{34}	s_3
	d_1	d_2	d_3	d_4	

Contoh

Pertimbangkan masalah dengan tiga pemasok yang memiliki persediaan 8, 10, dan 9, dan empat konsumen dengan permintaan masing-masing 6, 5, 8, dan 8. Solusi layak dasar dapat dengan mudah dibangun menggunakan metode serakah: mulai dari sudut kiri atas tabel dan alokasikan sebanyak mungkin ke setiap sel hingga kuota baris atau kolom terpenuhi. Tabel awal yang terisi mungkin terlihat seperti berikut:



x_{11}	5	x_{12}	3		4		6	8
		x_{22}	7	x_{23}	4		1	10
	5		6	x_{33}	2	x_{34}	4	9
	6		5		8		8	



Kita amati bahwa solusi layak dasar sesuai dengan pohon rentang dalam tabel transportasi. Secara umum, jika ada n pemasok dan m konsumen, jumlah total simpul adalah $n + m$, dan solusi layak dasar akan melibatkan $n + m - 1$ tepat variabel dasar (entri bukan nol), membentuk struktur pohon rentang. Ini sesuai dengan jumlah kendala ganda independen dalam sistem. Untuk menentukan nilai pengali Lagrange, kita secara sewenang-wenang menetapkan salah satunya - misalnya, kita tetapkan $\lambda_1 = 0$. Untuk setiap variabel dasar (yaitu, sisi dalam pohon rentang) di mana $x_{ij} \neq 0$, kelonggaran komplementer membutuhkan:

$$(c_{ij} - \lambda_i + \mu_j)x_{ij} = 0 \Rightarrow c_{ij} = \lambda_i - \mu_j$$

Dengan menggunakan hubungan ini dan biaya yang diketahui untuk sel yang ditempati, kita menyebarkan dan menghitung pengali Lagrange yang tersisa. Misalnya, jika $c_{11} = 5$ dan $\lambda_1 = 0$, maka $\mu_1 = -5$. Dengan melanjutkan cara ini, kita mengisi nilai pengali untuk semua pemasok dan konsumen:

	-5	-3	0	-2				
$\lambda_1 = 0$	6	5	2	3	4	6		
$\lambda_2 = 4$		2	3	7	7	4	1	
$\lambda_3 = 2$		5		6	1	2	8	4

Untuk memverifikasi kelayakan ganda, kita menghitung $c_{ij} - \lambda_i + \mu_j$ (yaitu, biaya tereduksi) untuk setiap sel yang tidak ditempati. Nilai-nilai ini harus nonnegatif agar solusi optimal. Kita menyusunnya ke dalam tabel:

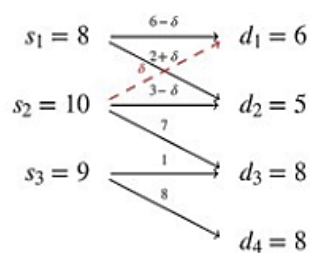


	-5	-3	0	-2			
$\lambda_1 = 0$	6	5	2	3	4	6	
$\lambda_2 = 4$	9	2	3	7	7	4	1
$\lambda_3 = 2$	7	5	6	1	2	8	4

Jika semua biaya tereduksi bernilai nonnegatif, solusi saat ini optimal. Namun, jika ada entri yang melanggar kondisi kelayakan ganda (yaitu, $c_{ij} - \lambda_i + \mu_j < 0$), solusi dapat ditingkatkan.

Untuk meningkatkan solusi, kita memilih sel dengan biaya tereduksi negatif - misalnya, sel yang menghubungkan pemasok kedua ke konsumen pertama - dan untuk sementara memasukkannya ke dalam basis. Ini menciptakan lingkaran tertutup atau siklus di antara sel-sel yang terisi. Kemudian kita menyesuaikan aliran di sepanjang siklus ini menggunakan metode *stepping-stone*, meningkatkan dan mengurangi alokasi secara bergantian untuk mempertahankan kelayakan. Siklus pada tabel transportasi bekerja seperti jalur penyesuaian. Satu sel dinaikkan, sel berikutnya diturunkan, lalu pola itu diulang agar jumlah pada setiap baris dan kolom tetap seimbang. Misalkan kita meningkatkan aliran pada tepi baru sebesar δ . Tabel yang diperbarui menjadi:

$6 - \delta$	5	$2 + \delta$	3		4		6
δ	2	$3 - \delta$	7	7	4		1
	5		6	1	2	8	4



Nilai maksimum yang diperbolehkan untuk δ ditentukan oleh aliran terkecil di antara aliran-aliran yang dikurangi. Dalam kasus ini, nilainya adalah $\delta = 3$. Penerapan pembaruan ini menghasilkan solusi layak yang baru:

3	5	5	3		4		6
3	2		7	7	4		1
	5		6	1	2	8	4



Sekarang kita menghitung ulang pengali Lagrange berdasarkan solusi layak dasar yang telah diperbarui:

	-5		-3		-7		-9	
$\lambda_1 = 0$	3	5	5	3	7	4	9	6
$\lambda_2 = -3$	3	2		7	7	4		1
$\lambda_3 = -5$		5		6	1	2	8	4

Kita mengamati pelanggaran kondisi kelayakan ganda pada sel kanan bawah. Oleh karena itu, diperlukan iterasi lain.

Kita memasukkan sel non-dasar yang melanggar ke dalam basis, yang membentuk lingkaran tertutup (siklus). Nilai aliran sepanjang siklus ini disesuaikan untuk mempertahankan kelayakan, meningkat dan menurun secara bergantian. Misalkan δ mewakili jumlah aliran yang akan ditambahkan ke sel yang masuk. Tabel yang diperbarui adalah:

3	5	5	3		4		6
3	2		7	$7 - \delta$	4	δ	1
	5		6	$1 + \delta$	2	$8 - \delta$	4

Nilai maksimum yang diperbolehkan adalah 7, setelah itu salah satu sel yang disesuaikan menjadi nol dan keluar dari basis. Menerapkan pembaruan ini menghasilkan tabel baru:

3	5	5	3		4		6
3	2		7		4	7	1
	5		6	8	2	1	4

Sekarang kita menghitung ulang pengali Lagrange berdasarkan solusi yang telah diperbarui ini:

	-5		-3		-2		-4	
$\lambda_1 = 0$	3	5	5	3	2	4		6
$\lambda_2 = -3$	3	2		7	0	-1		1
$\lambda_3 = 0$		5		6	8	2	1	4



Pada tahap ini, semua biaya tereduksi memenuhi kondisi kelayakan ganda, yang menunjukkan bahwa solusi saat ini optimal.

Masalah Aliran Maksimum

Subbab ini dapat dibaca sebagai cara mencari kapasitas terbaik sebuah jaringan. Bayangkan pipa air: setiap pipa punya batas tampung, dan tugas algoritma adalah mencari berapa banyak aliran yang masih bisa lewat tanpa melanggar batas itu. Pertimbangkan jaringan terarah $G = (V, E)$ dengan simpul sumber 1 dan simpul tujuan n . Setiap $(i, j) \in E$ sisi dikaitkan dengan kapasitas nonnegatif C_{ij} , dan tidak ada biaya transportasi. Tujuannya adalah untuk menentukan jumlah aliran maksimum yang dapat dikirim dari sumber ke tujuan, dengan mempertimbangkan kendala kapasitas dan konservasi aliran.

Meskipun ini dapat dirumuskan ulang sebagai kasus khusus dari masalah aliran biaya minimum dengan memperkenalkan sisi balik dari n ke 1 dengan kapasitas tak terbatas dan biaya -1 , sehingga mengubah masalah maksimisasi menjadi minimisasi biaya, metode yang lebih efisien dan intuitif ada khusus untuk menyelesaikan masalah aliran maksimum - terutama, yang didasarkan pada *Teorema Aliran Maksimum Potongan Minimum*. Masalah aliran maksimum dapat dirumuskan sebagai program linier berikut:

maksimalkan δ dengan batasan

$$\sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} = \begin{cases} \delta, & \text{if } i = 1, \\ -\delta, & \text{if } i = n, \\ 0, & \text{otherwise,} \end{cases} \quad \forall i \in V,$$

$$0 \leq x_{ij} \leq C_{ij} \quad \forall (i, j) \in E.$$

di mana x_{ij} menunjukkan aliran pada tepi (i, j) dan δ mewakili total aliran dari sumber ke tujuan. Untuk menganalisis masalah ini, kita memerlukan gagasan tentang potongan dalam jaringan aliran.

Definisi 8.19 (Potongan/Cut)

Misalkan $G = (V, E)$ adalah sebuah jaringan aliran dengan kapasitas C_{ij} untuk setiap $(i, j) \in E$. Sebuah *potongan* adalah partisi dari himpunan titik V ke dalam dua subhimpunan lepas S dan $V \setminus S$, sedemikian sehingga sumber $1 \in S$ dan muara $n \in V \setminus S$. Kapasitas dari *potongan* $(S, V \setminus S)$ didefinisikan sebagai:

$$C(S) = \sum_{(i,j) \in (S \times (V \setminus S)) \cap E} C_{ij}$$

Misalkan x adalah vektor aliran yang layak yang mengirimkan δ unit dari simpul 1 ke simpul n . Untuk himpunan bagian $X, Y \subseteq V$, definisikan aliran total dari X ke Y sebagai:

$$f_x(X, Y) = \sum_{(i,j) \in (X \times Y) \cap E} x_{ij}$$



Sekarang, perhatikan potongan apa pun $(S, V \setminus S)$ dengan $1 \in S$ dan $n \in V \setminus S$. Nilai alirannya dapat dinyatakan sebagai:

$$\delta = \sum_{i \in S} \left(\sum_{j: (i,j) \in E} x_{ij} - \sum_{j: (j,i) \in E} x_{ji} \right)$$

yang diperoleh dari kendala konservasi aliran: semua suku dalam penjumlahan bernilai nol kecuali ketika $i = 1$, di mana aliran keluar bersih sama dengan δ .

Dengan menulis ulang, kita peroleh:

$$\begin{aligned} \delta &= f_x(S, V) - f_x(V, S) \\ &= f_x(S, V \setminus S) - f_x(V \setminus S, S) \\ &\leq f_x(S, V \setminus S) \\ &\leq C(S), \end{aligned}$$

di mana ketidaksetaraan terakhir berlaku karena aliran pada setiap sisi tidak dapat melebihi kapasitasnya.

Penurunan ini menetapkan bahwa nilai aliran yang layak dibatasi di atas oleh kapasitas potongan mana pun. Wawasan luar biasa dari *Teorema Aliran Maksimum Potongan Minimum* adalah bahwa batas atas ini ketat: selalu ada potongan di mana nilai aliran maksimum sama dengan kapasitas potongan tersebut $C(S)$.

Teorema 8.7 (Teorema Aliran Maksimum Potongan Minimum)

Subbab ini dapat dibaca sebagai cara mencari kapasitas terbaik sebuah jaringan. Bayangkan pipa air: setiap pipa punya batas tampung, dan tugas algoritma adalah mencari berapa banyak aliran yang masih bisa lewat tanpa melanggar batas itu. *Misalkan δ adalah nilai aliran maksimum dalam jaringan. Maka*

$$\delta = \min\{C(S) : S \subseteq V, 1 \in S, n \in V \setminus S\}$$

di mana $C(S)$ adalah kapasitas potongan $(S, V \setminus S)$.

Bukti:

Misalkan x adalah aliran layak. Definisikan *jalur penambah* sebagai urutan simpul $v_0, \dots, v_k$ sedemikian sehingga untuk setiap $i = 1, \dots, k$, salah satu kondisi berikut berlaku:

$$x_{v_{i-1}v_i} < C_{v_{i-1}v_i} \text{ (sisi maju dengan kapasitas tersedia),}$$

Atau

$$x_{v_i v_{i-1}} > 0 \text{ (sisi mundur dengan aliran positif).}$$

Jalur seperti itu memungkinkan kita meningkatkan aliran pada sisi maju dan/atau mengurangi aliran pada sisi mundur sehingga meningkatkan aliran total δ .

Sekarang, anggaplah x adalah aliran maksimum. Definisikan himpunan:



$$S = \{1\} \cup \{i \in V \mid \text{terdapat jalur penambah dari } 1 \text{ ke } i\}$$

Karena x sudah optimal, tidak mungkin ada *jalur penambah* dari 1 ke n ; oleh karena itu, $n \in V \setminus S$. Dari hasil sebelumnya, nilai aliran diberikan oleh:

$$\delta = f_x(S, V \setminus S) - f_x(V \setminus S, S)$$

di mana $f_x(A, B)$ menyatakan total aliran dari simpul dalam himpunan A ke simpul dalam himpunan B . Kita mengklaim bahwa $f_x(V \setminus S, S) = 0$. Jika tidak, terdapat sisi (v, u) dengan $v \in V \setminus S$, $u \in S$, dan $x_{vu} > 0$. Menambahkan (v, u) ke *jalur penambah* menuju u akan membentuk *jalur penambah* menuju v , yang bertentangan dengan definisi S .

Selanjutnya, kita tegaskan bahwa $f_x(S, V \setminus S) = C(S)$. Jika tidak, terdapat sisi $(i, j) \in (S \times (V \setminus S))$ sedemikian sehingga $x_{ij} < C_{ij}$, yang memungkinkan *jalur penambah* menuju $j \in V \setminus S$. Sekali lagi, hal ini bertentangan dengan definisi S . Oleh karena itu,

$$\delta = f_x(S, V \setminus S) = C(S).$$

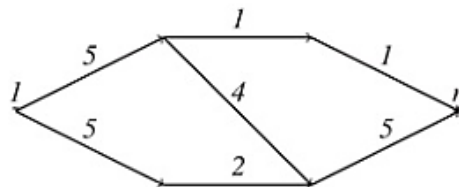
Teorema Aliran Maksimum Potongan Minimum memberikan jaminan optimalitas tetapi tidak secara langsung menghasilkan algoritma untuk menemukan aliran maksimum. Namun, pendekatan sederhana dan efektif untuk memperoleh solusi optimal disediakan oleh *Algoritma Ford–Fulkerson*, yang berulang kali meningkatkan aliran di sepanjang jalur yang tersedia dari sumber ke tujuan.

Ford–Fulkerson

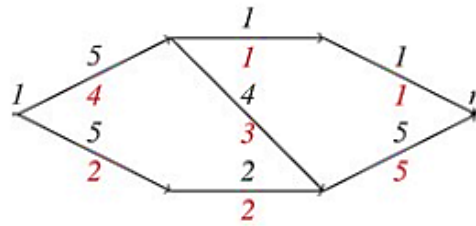
- a. Inisialisasi aliran $x_{ij} = 0$ untuk semua $(i, j) \in E$
- b. Selama terdapat *jalur penambah* dari sumber 1 ke tujuan dalam grafik residual:
 - i. Tentukan kapasitas residual minimum di sepanjang jalur tersebut.
 - ii. Tambahkan aliran di sepanjang jalur tersebut sebanyak satuan.
- c. Ketika tidak ada lagi *jalur penambah*, aliran saat ini optimal.

Contoh 8.4

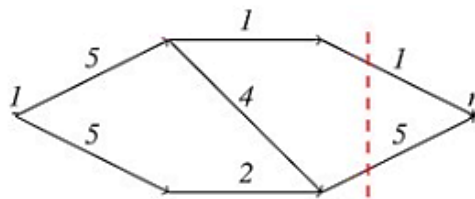
Perhatikan jaringan aliran yang diilustrasikan di bawah ini:



Dengan menerapkan algoritma Ford–Fulkerson, kita berulang kali meningkatkan aliran hingga mencapai konfigurasi berikut:



Nilai aliran total adalah $\delta = 6$ Dengan mengidentifikasi potongan (ditunjukkan di bawah dengan garis putus-putus) dengan kapasitas 6, kita mengkonfirmasi dengan Teorema Aliran Maksimum Potongan Minimum bahwa solusi ini optimal:



8.6 LATIHAN

1. **Perumusan Pemrograman Linier dan Solusi Grafis:** Sebuah pabrik kecil memproduksi dua jenis sensor pintar, S_1 dan S_2 .
 - Memproduksi satu unit membutuhkan waktu S_1 perakitan 2 jam dan waktu pengujian 1 jam. Memproduksi satu unit membutuhkan waktu S_2 perakitan 1 jam dan waktu pengujian 3 jam. Total waktu perakitan yang tersedia adalah 10 jam per minggu.
 - Total waktu pengujian yang tersedia adalah 15 jam per minggu.
 - Keuntungan untuk setiap unit S_1 adalah Rp. 50.000, dan untuk setiap unit S_2 adalah Rp. 40.000.
 - i. Perumusan: Definisikan variabel keputusan dan rumuskan masalah ini sebagai program linier (LP) untuk memaksimalkan total keuntungan.
 - ii. Solusi Grafis: Plot kendala untuk menentukan wilayah yang layak. Identifikasi secara grafis solusi optimal yang memaksimalkan fungsi tujuan (x_1, x_2) dan nyatakan keuntungan maksimum.
2. **Kekonveksian dan Matriks Hessian:** Dalam optimasi konveks, kondisi kunci agar suatu fungsi $f(x)$ menjadi konveks adalah matriks H Hessian-nya harus positif semidefinit. Keamanan kriptografi bergantung pada masalah matematika yang mudah dihitung satu arah, tetapi sangat sulit dibalik tanpa informasi rahasia. Inilah alasan kunci dan asumsi komputasi perlu dijelaskan dengan jelas. Pertimbangkan fungsi $f(x_1, x_2) = x_1^2 + 3x_1x_2 + 2x_2^2 + 5x_1 + x_2$.
 1. Perhitungan Hessian: Hitung matriks H Hessian untuk fungsi $f(x_1, x_2)$.



2. Uji Kekonveksian: Dengan menggunakan definisi bahwa suatu matriks H bersifat semidefinit positif jika $v^T H v \geq 0$ untuk semua vektor (atau dengan memeriksa minor utama), tentukan apakah fungsi $f(x_1, x_2)$ konveks.
3. **Analisis Metode Optimasi Iteratif:** Penyelesaian sistem persamaan linear dapat dirumuskan sebagai meminimalkan fungsi kuadrat $f(x) = \frac{1}{2} x^T A x - b^T x + c$. Laju konvergensi metode iteratif seperti Penurunan Tercuram bergantung pada bilangan kondisi, $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$.
 1. Ukuran Langkah Penurunan Tercuram: Jelaskan tujuan ukuran langkah optimal dalam metode Penurunan Tercuram, dan tuliskan rumus yang digunakan untuk menghitung ukuran langkah optimal ini untuk persamaan $x_{i+1} = x_i + \alpha_i r_i$.
 2. Perbandingan Konvergensi: Bandingkan laju konvergensi metode Penurunan Tercuram dengan metode Gradien Konjugasi dalam hal bilangan kondisi κ , dan jelaskan implikasi praktis dari perbedaan ini untuk menyelesaikan sistem besar.
4. **Dualitas Pemrograman Linier dan Kelonggaran Komplementer:** Pertimbangkan Program Linier Primal (LP) dalam bentuk umum:
$$\begin{array}{ll}\text{Minimize} & c^T x \\ \text{Subject to} & Ax \geq b, x \geq 0\end{array}$$
 1. Formulasi Ganda: Nyatakan Program Linier Ganda (LP) yang sesuai.
 2. Kelonggaran Komplementer: Nyatakan dua kondisi untuk kelonggaran komplementer untuk pasangan solusi primal-dual optimal (x, y) . Jelaskan secara sederhana apa yang diungkapkan oleh kondisi pertama, $y_i(Ax - b)_i = 0$, tentang hubungan antara variabel primal optimal dan kendala.
5. **Teorema Aliran Maksimum dan Potongan Minimum:** Masalah Aliran Maksimum dalam jaringan bertujuan untuk menemukan aliran maksimum yang mungkin dari sumber ke tujuan dengan mempertimbangkan kapasitas tepi. Teorema Aliran Maksimum-Potongan Minimum memberikan prinsip dasar untuk masalah ini.
 1. Pernyataan Teorema: Nyatakan *Teorema Aliran Maksimum-Potongan Minimum*.
 2. Algoritma dan Optimalitas: Jelaskan bagaimana algoritma *Ford–Fulkerson* digunakan untuk menemukan aliran maksimum. Jelaskan sifat penentu dari himpunan simpul (potongan) yang dihasilkan ketika algoritma *Ford–Fulkerson* berakhir dengan aliran maksimum dan tunjukkan bagaimana sifat ini menghubungkan nilai aliran maksimum dengan kapasitas potongan minimum.



BAB 9

ALGORITMA PENGAMBILAN KEPUTUSAN

9.1 PROSES KEPUTUSAN MARKOV

GAM menggabungkan beberapa fungsi halus dari fitur yang berbeda. Hasilnya lebih fleksibel daripada regresi linier biasa, tetapi masih cukup mudah dijelaskan. Dalam lingkungan yang kompleks seperti energi, lingkungan, dan pengelolaan sumber daya, proses pengambilan keputusan sering kali melibatkan ketidakpastian, berbagai tujuan, dan kondisi yang berubah secara dinamis. Bab ini memperkenalkan kerangka kerja dan model algoritmik yang dirancang untuk membantu pengambilan keputusan dalam konteks tersebut. Dimulai dengan dasar teoritis proses keputusan Markov (MDP), bab ini diperluas ke metode praktis seperti pembelajaran penguatan (RL), *iterasi nilai*, pembelajaran Q, dan pembelajaran perbedaan temporal (TD). Algoritma ini sangat berharga dalam sistem otonom, pemantauan lingkungan, dan jaringan energi adaptif, di mana agen harus mempelajari strategi optimal melalui interaksi dengan lingkungan. Pada akhir bab ini, pembaca akan memahami bagaimana model pengambilan keputusan beroperasi, bagaimana kebijakan dioptimalkan dari waktu ke waktu, dan bagaimana menerapkan teknik ini dalam skenario dunia nyata. Setelah membaca bab ini, Anda seharusnya dapat:

- Menjelaskan komponen fundamental dari kerangka kerja pengambilan keputusan seperti MDP dan RL. Mengidentifikasi model keputusan yang tepat untuk sistem infrastruktur lingkungan dan energi.
- Membedakan antara kebijakan deterministik dan stokastik, serta antara pendekatan pembelajaran berbasis model dan tanpa model.
- Menerapkan algoritma seperti *iterasi nilai*, *pembelajaran-Q*, dan *pembelajaran TD* untuk menyelesaikan masalah pengambilan keputusan dalam kondisi ketidakpastian.
- Menginterpretasikan peran fungsi penghargaan, faktor diskon, dan konvergensi kebijakan dalam perencanaan jangka panjang.

Pembaca dapat memandang keadaan sebagai foto kondisi sistem saat ini. Aksi adalah pilihan yang tersedia, sedangkan transisi menunjukkan kemungkinan perubahan setelah aksi itu dipilih. MDP membantu kita menyusun keputusan berurutan. Cara berpikirnya sederhana: lihat keadaan saat ini, pilih aksi, terima imbalan, lalu sistem bergerak ke keadaan berikutnya. Misalkan $N = \{0, 1, \dots\}$, dan misalkan $\mathbb{R}^n$ menyatakan ruang Euclidean berdimensi n , dengan $\mathbb{R} = \mathbb{R}^1$. Sebuah **MDP** ditentukan oleh komponen-komponen berikut:

- *Ruang keadaan* X
- *Ruang aksi* A
- Seperangkat *aksi yang tersedia* $A(x)$ untuk setiap keadaan $x \in X$
- Seperangkat *probabilitas transisi* $p(Y | x, a)$, yang menggambarkan *probabilitas transisi* ke suatu himpunan $Y \subseteq X$ dari keadaan x setelah mengambil aksi a
- *Fungsi imbalan* $r(x, a)$, yang mendefinisikan imbalan satu langkah untuk mengeksekusi aksi a di keadaan x



Komponen-komponen ini mendefinisikan sistem stokastik di mana, pada setiap langkah, sistem berada dalam keadaan $x \in X$, dan pengambil keputusan memilih suatu aksi $a \in A(x)$. Sistem kemudian bertransisi ke keadaan baru sesuai dengan distribusi probabilitas $p(\cdot | x, a)$, dan pengambil keputusan menerima imbalan langsung $r(x, a)$. Pilihan aksi dapat bergantung pada keadaan saat ini, langkah waktu, atau riwayat sistem. Aksi dapat dipilih secara deterministik (tidak diacak) atau probabilistik (diacak) dari himpunan $A(x)$.

MDP Diskrit

Pada MDP diskrit, keadaan dan aksi dapat dihitung satu per satu. Bentuk ini cocok untuk belajar konsep dasar sebelum masuk ke sistem yang *ruang keadaannya* lebih luas. Suatu MDP dikatakan *terbatas* jika himpunan keadaan dan himpunan aksinya terbatas. Disebut *diskrit* jika himpunan-himpunan ini terbatas atau tak terhingga terhitung. MDP diskrit merupakan bidang utama pengembangan teoritis dan terapan. Dalam kasus ini, tidak diperlukan asumsi terukur tingkat lanjut untuk elemen-elemen kunci model. Akibatnya, bahkan pembaca tanpa latar belakang teori ukuran dapat memahami sebagian besar literatur tentang MDP diskrit. Beberapa hasil yang melibatkan pengaturan yang lebih umum mungkin dibatasi pada *ruang keadaan* dan aksi diskrit.

Dalam *ruang keadaan* diskrit X , *probabilitas transisi* biasanya ditulis sebagai $p(y | x, a)$ atau $p_{xy}(a)$, di mana $x, y \in X$. Notasi tambahan seperti i, j, k juga dapat mewakili keadaan. Kecuali dinyatakan lain, kita mengasumsikan $p(X | x, a) = 1$, yang berarti sistem selalu bertransisi ke beberapa keadaan di dalam X . Waktu diindeks dengan t, s atau $n \in N$. Trajektori adalah urutan dalam bentuk $x_0 a_0 x_1 a_1 \dots$, dan himpunan semua kemungkinan *trajektori* tak terhingga dilambangkan dengan $H_\infty = (X \times A)^\infty$. *Sejarah* dengan panjang n ditulis sebagai $h_n = x_0 a_0 \dots x_{n-1} a_{n-1} x_n$, dan himpunan sejarah tersebut adalah $H_n = X \times (A \times X)^n$.

Kebijakan non-acak didefinisikan sebagai urutan pemetaan $\phi_n: H_n \rightarrow A$, untuk $n \in N$, sedemikian sehingga $\phi_n(h_n) \in A(x_n)$ untuk setiap sejarah h_n . Jika kebijakan hanya bergantung pada keadaan saat ini x_n , yaitu, $\phi_n(x_n)$, maka disebut *kebijakan Markov*. Dengan demikian, kebijakan Markov terdiri dari pemetaan yang $\phi_n: X \rightarrow A$ memenuhi $\phi_n(x) \in A(x)$ untuk semua $x \in X$ dan untuk setiap n . Jika ϕ_n sama untuk semua n , kebijakan tersebut disebut *stasioner*, dan didefinisikan oleh satu pemetaan $\phi: X \rightarrow A$ dengan $\phi(x) \in A(x)$ untuk semua $x \in X$. Kita menunjuk kelas kebijakan sebagai berikut:

- PS : himpunan semua *kebijakan non-acak*
- PS^M : himpunan semua *kebijakan Markov*
- PS^S : himpunan semua *kebijakan stasioner*

Ini memenuhi relasi inklusi: $PS^S \subseteq PS^M \subseteq PS$. Seperti yang telah dibahas sebelumnya, mengizinkan pengacakan dalam pemilihan tindakan memperluas himpunan kebijakan yang tersedia. *Kebijakan acak* π didefinisikan sebagai urutan probabilitas transisi $\pi_n(a_n | h_n)$ dari ruang sejarah H_n ke ruang tindakan A , untuk setiap $n \in N$. Probabilitas ini harus memenuhi kondisi berikut:

$$\pi_n(A(x_n) | x_0 a_0 \dots x_{n-1} a_{n-1} x_n) = 1,$$



memastikan bahwa tindakan yang dipilih termasuk dalam himpunan yang layak. Jika probabilitas tindakan hanya bergantung pada keadaan saat ini x_n dan bukan pada seluruh riwayat yaitu,

$$\pi_n(a_n | x_0 a_0 \cdots x_{n-1} a_{n-1} x_n) = \pi_n(a_n | x_n),$$

Maka kebijakan tersebut disebut *kebijakan Markov acak*.

Lebih lanjut, jika distribusi probabilitas konsisten di semua langkah waktu, yaitu, $\pi_n(\cdot | x) = \pi_m(\cdot | x)$ untuk semua $n, m \in N$, kebijakan tersebut disebut *kebijakan stasioner acak*. Kebijakan semacam itu didefinisikan oleh pemetaan probabilitas transisi tunggal yang $\pi(a | x)$ memenuhi $\pi(A(x) | x) = 1$ untuk setiap $x \in X$. Kita mengkategorikan kebijakan-kebijakan ini sebagai berikut:

- PS^R : himpunan semua *kebijakan acak*
- PS^{RM} : himpunan *kebijakan Markov acak*
- PS^{RS} : himpunan *kebijakan stasioner acak*

Himpunan-himpunan ini memenuhi inklusi bersarang:

$$PS^{RS} \subseteq PS^{RM} \subseteq PS^R,$$

dan hal tersebut juga berkaitan dengan kelas *kebijakan yang tidak diacak* sebagai berikut:

$$PS^S \subseteq PS^{RS}, \quad PS^M \subseteq PS^{RM}, \quad PS \subseteq PS^R.$$

Penting untuk dicatat bahwa terminologi dapat bervariasi di seluruh literatur. Secara khusus, tidak ada konsensus universal tentang apakah istilah "stasioner" menyiratkan perilaku yang tidak diacak atau apakah kebijakan diasumsikan diacak secara default. Terminologi alternatif yang umum meliputi:

- *Kebijakan Murni*: sinonim untuk *kebijakan yang tidak diacak*
- *Kebijakan Deterministik*: biasanya mengacu pada *kebijakan stasioner* yang tidak diacak

Sistem yang diatur oleh suatu kebijakan berevolusi sebagai berikut: pada langkah waktu n , jika proses berada dalam keadaan x dengan riwayat h_n , suatu tindakan a dipilih (berpotensi secara acak) sesuai dengan kebijakan π . Sistem kemudian bertransisi ke keadaan baru y pada waktu $n + 1$ dengan probabilitas $p(y | x, a)$.

Diberikan keadaan x awal dan kebijakan π , proses stokastik ini menghasilkan urutan $x_0 a_0 x_1 a_1 \cdots$, yang distribusi berdimensi terbatasnya ditentukan oleh aturan di atas. Dengan teorema perluasan Kolmogorov, proses ini terdefinisi dengan baik, dan kita menunjukkan ukuran probabilitas dan ekspektasi yang $\mathbb{P}_x^\pi$ dihasilkan $\mathbb{E}_x^\pi$, masing-masing dengan π . Perlu dicatat, $\mathbb{P}_x^\pi\{x_0 = x\} = 1$, memastikan bahwa kondisi awal terpenuhi. Kebijakan stasioner apa pun ϕ menginduksi rantai Markov pada *ruang keadaan* X , dengan probabilitas transisi yang diberikan oleh

$$p_{xy}(\phi) = p(y | x, \phi(x)).$$



Demikian pula, kebijakan π stasioner acak juga mendefinisikan rantai Markov di atas X , tetapi dalam kasus ini, *probabilitas transisi* dihitung sebagai

$$p_{xy}(\pi) = \sum_{a \in A(x)} \pi(a) p(y | x, a).$$

Matriks $P(\pi)$ tersebut menunjukkan matriks transisi $p_{xy}(\pi)$ yang sesuai dengan elemen-elemennya. Distribusi batas dijelaskan oleh matriks tersebut.

$$Q(\pi) = \liminf_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} P^n(\phi), \quad (9.1)$$

yang selalu ada dan bersifat stokastik asalkan ruang X keadaan terbatas.

Misalkan f adalah *fungsi imbalan terminal*, dan misalkan β adalah *faktor diskon*. Total imbalan yang diharapkan selama langkah-langkah pertama N , di bawah kebijakan π yang dimulai dari keadaan x , diberikan oleh:

$$v_N(x, \pi, \beta, f) = \mathbb{E}_x^\pi \left[\sum_{n=0}^{N-1} \beta^n r(x_n, a_n) + \beta^N f(x_N) \right], \quad (9.2)$$

asalkan harapan ini didefinisikan dengan baik.

Jika $\beta \in [0,1)$, ekspresi tersebut mewakili *total imbalan yang diharapkan setelah didiskon*. Ketika $\beta = 1$, itu sesuai dengan *total imbalan yang diharapkan tanpa diskon*, juga dikenal sebagai *total imbalan*. Ketika faktor diskon tetap $\beta \in [0,1)$, kita sering menyingkat notasi sebagai $v(x, \pi)$ sebagai pengganti bentuk lengkapnya $v(x, \pi, \beta)$. Total imbalan yang diharapkan selama horizon tak terbatas didefinisikan sebagai:

$$v(x, \pi) = v(x, \pi, \beta) = v_\infty(x, \pi, \beta, 0), \quad (9.3)$$

di mana $\beta \in [0,1)$ *faktor diskonnya*. Perhatikan bahwa memastikan keberadaan total hadiah $v(x, \pi, 1)$ (yaitu, untuk kasus tanpa diskon) biasanya membutuhkan kondisi yang lebih kuat daripada yang dibutuhkan untuk hadiah $v(x, \pi, \beta)$ yang didiskon ketika $0 \leq \beta < 1$. *Rata-rata hadiah yang diharapkan per langkah waktu* di bawah kebijakan didefinisikan oleh:

$$w(x, \pi) = \liminf_{N \rightarrow \infty} \frac{1}{N} v_N(x, \pi, 1, 0). \quad (9.4)$$

Misalkan kita memiliki kriteria kinerja yang didefinisikan untuk semua kebijakan. Maka, *fungsi nilai kinerja optimal* diberikan oleh:

$$G(x) = \sup_{\pi \in PS^R} g(x, \pi), \quad (9.5)$$



di mana PS^R menunjukkan himpunan semua kebijakan yang diacak. Sesuai dengan itu, fungsi nilai optimal di bawah model penghargaan yang berbeda adalah

$$V_N(x, \beta, f) = \sup_{\pi \in PS^R} v_N(x, \pi, \beta, f), \quad (9.6)$$

$$V(x) = V(x, \beta) = \sup_{\pi \in PS^R} v(x, \pi, \beta), \quad (9.7)$$

$$W(x) = \sup_{\pi \in PS^R} w(x, \pi). \quad (9.8)$$

Suatu kebijakan π dikatakan ϵ -*optimal* terhadap kriteria kinerja g jika memenuhi:

$$g(x, \pi) \geq G(x) - \epsilon \quad \text{for all } x \in X,$$

di mana $\epsilon \geq 0$. Dalam kasus khusus di mana $\epsilon = 0$, kebijakan tersebut hanya disebut sebagai *optimal*. Diberikan fungsi yang f didefinisikan pada ruang keadaan X , kita mendefinisikan operator berikut yang terkait dengan imbalan dan optimasi:

Operator Imbalan

Operator imbalan memisahkan hasil langsung dari konsekuensi masa depan. Dengan pemisahan ini, pembaca bisa melihat bagian mana yang berasal dari keputusan sekarang dan bagian mana yang berasal dari nilai lanjutan.

$$P^a f(x) = \mathbb{E}[f(x_1) \mid x_0 = x, a_0 = a], \quad (9.9)$$

$$T_\beta^a f(x) = r(x, a) + \beta P^a f(x), \quad (9.10)$$

di mana $P^a f(x)$ mewakili nilai yang diharapkan dari f pada keadaan berikutnya jika tindakan a diambil pada keadaan x , dan $T_\beta^a f(x)$ mewakili total imbalan yang diharapkan untuk mengambil tindakan a pada keadaan x , dengan memperhitungkan baik imbalan langsung maupun imbalan masa depan yang didiskon.

Operator Optimalitas

Operator optimalitas menambahkan langkah memilih aksi terbaik. Jadi, model tidak hanya menilai satu kebijakan, tetapi juga mencari kebijakan yang memberikan nilai paling tinggi.

$$Pf(x) = \sup_{a \in A(x)} P^a f(x), \quad (9.11)$$

$$T_\beta f(x) = \sup_{a \in A(x)} T_\beta^a f(x), \quad (9.12)$$

di mana supremum diambil atas semua tindakan yang layak pada keadaan x , menghasilkan nilai harapan terbaik atau imbalan total terbaik untuk keadaan tersebut. *Persamaan optimalitas horizon terbatas* kemudian diberikan oleh relasi rekursif:



$$V_{N+1}(x) = T_{\beta}V_N(x), \quad \text{for } N = 0, 1, \dots \quad (9.13)$$

Persamaan ini menjelaskan bagaimana fungsi nilai diperbarui dari langkah N ke langkah $N + 1$ menggunakan operator optimalitas T_{β} . *Persamaan optimalitas imbalan* terdiskon didefinisikan sebagai:

$$V(x) = T_{\beta}V(x), \quad (9.14)$$

di mana T_{β} adalah operator optimalitas terdiskonto dan $V(x)$ mewakili fungsi nilai di bawah optimasi kebijakan. Suatu tindakan $a \in A(x)$ dikatakan konservatif pada keadaan $(N + 1)$ untuk masalah -langkah jika memenuhi:

$$T_{\beta}^a V_N(x) = T_{\beta}V_N(x).$$

Demikian pula, suatu tindakan $a \in A(x)$ bersifat konservasi pada keadaan x untuk masalah diskon horizon tak terbatas jika:

$$T_{\beta}^a V(x) = T_{\beta}V(x).$$

Dalam kasus khusus di mana $\beta = 1$, kita nyatakan $T^a = T_1^a$ dan $T = T_1$. *Persamaan optimalitas imbalan total yang tidak didiskon* yang sesuai menjadi:

$$V(x) = TV(x), \quad (9.15)$$

yang menyatakan kondisi optimalitas untuk imbalan total (tanpa diskon) yang diharapkan.

Untuk masalah imbalan total, fungsi nilai biasanya memenuhi persamaan optimalitas di atas. Selain itu, himpunan tindakan konservasi pada setiap langkah $n = 1, \dots, N + 1$ bertepatan dengan himpunan tindakan optimal untuk masalah keputusan $(N + 1)$ -langkah yang sesuai. Di bawah kondisi teknis tambahan, gabungan tindakan konservasi di seluruh langkah juga dapat mendefinisikan himpunan tindakan optimal untuk masalah horizon tak terbatas. Hasil ini akan diuraikan lebih lanjut di bab-bab selanjutnya. *Persamaan optimalitas untuk imbalan rata-rata* diberikan oleh:

$$W(x) = PW(x), \quad (9.16)$$

$$W(x) + h(x) = \sup_{a \in A'(x)} T^a h(x), \quad (9.17)$$

di mana himpunan tindakan konservasi didefinisikan sebagai:

$$A'(x) = \{a \in A(x) : P^a W(x) = PW(x)\}. \quad (9.18)$$



Persamaan (9.16) dikenal sebagai *persamaan optimalitas pertama*, sedangkan Persamaan (9.17) disebut sebagai *persamaan optimalitas kedua*. Jika fungsi imbalan $W(x)$ rata-rata konstan, yaitu, $W(x) = W$ untuk semua $x \in X$, maka persamaan pertama berlaku secara trivial, dan setiap tindakan menjadi konservatif, yaitu, $A'(x) = A(x)$. Dalam kasus khusus ini, persamaan kedua direduksi menjadi bentuk yang disederhanakan:

$$W + h(x) = Th(x), \quad (9.19)$$

yang umumnya dikenal sebagai *persamaan optimalitas imbalan rata-rata*. Dalam beberapa pengaturan, keadaan awal x dapat ditentukan oleh distribusi probabilitas μ pada ruang keadaan X . Dalam kasus tersebut, semua definisi dan notasi di atas tetap berlaku, dengan x digantikan oleh μ . Sebagai contoh, kita menuliskannya sebagai $\mathbb{P}_\mu^\pi$, $\mathbb{E}_\mu^\pi$, $v(\mu, \pi)$, $V(\mu)$, $w(\mu, \pi)$, dan $W(\mu)$.

Secara umum, optimalitas dan ϵ -optimalitas yang didefinisikan terhadap distribusi awal μ lebih kuat daripada yang didefinisikan untuk keadaan awal tetap x . Namun, dalam banyak situasi standar—seperti masalah imbalan total—kedua definisi ini ekuivalen. Formulasi yang lebih umum muncul ketika masalah pengambilan keputusan melibatkan berbagai tujuan. Misalkan terdapat $(K + 1)$ fungsi imbalan $r_k(x, a)$, untuk $k = 0, \dots, K$. Dalam pengaturan horizon berhingga, imbalan akhir juga dapat bergantung pada indeks k . Oleh karena itu, semua fungsi yang terkait diindeks dengan k , seperti:

$$w_k(x, \pi), \quad f_k(x), \quad \text{and} \quad W_k(x).$$

Ketika berurusan dengan berbagai tujuan, umumnya lebih mudah untuk menetapkan keadaan awal x . Namun, dimungkinkan juga untuk mendefinisikan masalah menggunakan distribusi awal μ pada ruang keadaan. Dalam kasus tersebut, notasi dan definisinya tetap sama, dengan μ menggantikan x secara keseluruhan. Untuk penyederhanaan, kita akan melanjutkan dengan asumsi bahwa kondisi awal diberikan oleh keadaan tetap x . Jika kinerja suatu kebijakan π dievaluasi menggunakan $(K + 1)$ kriteria $g_k(x, \pi)$ untuk $k = 0, \dots, K$, maka salah satu tujuan yang umum adalah memaksimalkan kriteria utama $g_0(x, \pi)$, dengan tunduk pada pemenuhan kendala-kendala pada K kriteria yang tersisa $g_1, \dots, g_K$. Misalkan C_k adalah batas bawah yang ditentukan untuk kendala-kendala ini, di mana $k = 1, \dots, K$.

$$g_k(x, \pi) \geq C_k, \quad \text{for all } k = 1, \dots, K. \quad (9.20)$$

Di antara semua kebijakan yang layak (feasible policies), sebuah kebijakan π disebut optimal untuk masalah optimasi dengan batasan jika ia secara tambahan memenuhi:

$$g_0(x, \pi) \geq g_0(x, \sigma), \quad \text{for every feasible policy } \sigma. \quad (9.21)$$



MDP Nondiskrit: Konstruksi Umum

Ketika ruang keadaan X atau ruang aksi A tidak diskrit, wajar untuk memodelkannya sebagai ruang terukur yang dilengkapi dengan σ -field yang sesuai, yang dilambangkan $\mathcal{X}$ dan $\mathcal{A}$, masing-masing. Dalam kasus di mana X atau A diskrit σ -field yang sesuai hanyalah himpunan kuasa (yaitu, himpunan semua himpunan bagian) dari ruang masing-masing. Kita juga mengasumsikan bahwa himpunan aksi yang layak $A(x)$ adalah himpunan bagian terukur dari $(A, \mathcal{A})$ untuk setiap $x \in X$. Kondisi ini secara otomatis terpenuhi ketika *ruang aksi* A diskrit. Kecuali dinyatakan lain, kita mengadopsi σ -field Borel standar $\mathcal{B}(\mathbb{R})$ pada $\mathbb{R}$, yang merupakan σ -field terkecil yang berisi semua interval terbuka. Untuk MDP dengan struktur nondiskrit, diasumsikan lebih lanjut bahwa:

- Fungsi hadiah terukur pada ruang produk $(X \times A, \mathcal{X} \times \mathcal{A})$.
- Probabilitas transisi $p(Y | x, a)$ adalah *probabilitas transisi reguler* dari $(X \times A, \mathcal{X} \times \mathcal{A})$ ke $(X, \mathcal{X})$.

Ingat bahwa fungsi transisi disebut *probabilitas transisi reguler* dari ruang $(E_1, \mathcal{E}_1)$ terukur ke ruang terukur $(E_2, \mathcal{E}_2)$ lainnya jika dua kondisi berikut terpenuhi:

- a. Untuk setiap $e_2 \in E_2$, fungsi tersebut $p(\cdot | e_2)$ merupakan ukuran probabilitas pada $(E_1, \mathcal{E}_1)$.
- b. Untuk setiap $B \in \mathcal{E}_1$, fungsi $p(B | \cdot)$ tersebut terukur pada $(E_2, \mathcal{E}_2)$.

Untuk mendefinisikan kebijakan dalam pengaturan umum (nondiskrit), kita memperkenalkan σ -bidang pada ruang sejarah. Secara khusus, untuk setiap $n \in \mathbb{N}$, misalkan ruang sejarah didefinisikan sebagai:

$$H_n = X \times (A \times X)^n,$$

dan misalkan kolom yang bersesuaian adalah:

$$\mathcal{H}_n = \mathcal{X} \times (\mathcal{A} \times \mathcal{X})^n.$$

Kebijakan non-acak dan acak dalam konteks ini didefinisikan secara analog dengan kebijakan pada kasus diskrit, dengan asumsi terukur standar tambahan:

- a. Kebijakan π non-acak diwakili oleh urutan pemetaan terukur $\pi_n: H_n \rightarrow A$, masing-masing terukur terhadap $(H_n, \mathcal{H}_n)$.
- b. Kebijakan Markov dan stasioner diberikan oleh pemetaan terukur yang didefinisikan pada $(X \times A, \mathcal{X} \times \mathcal{A})$.

Untuk *kebijakan acak*, kita mengasumsikan:

- Pemetaan tersebut π_n adalah *probabilitas transisi reguler* dari $(H_n, \mathcal{H}_n)$ ke $(A, \mathcal{A})$.
- Untuk *kebijakan Markov* dan stasioner acak, *probabilitas transisi* didefinisikan dari $(X \times A, \mathcal{X} \times \mathcal{A})$ ke $(A, \mathcal{A})$.

Biarkan ruang sejarah tak terbatas dilambangkan dengan:

$$\mathcal{H}_\infty = (\mathcal{X} \times \mathcal{A})^\infty$$



Setiap keadaan awal x dan kebijakan π secara unik menentukan ukuran probabilitas pada ruang terukur $(H_\infty, \mathcal{H}_\infty)$. Ukuran ini dinotasikan dengan $\mathbb{P}_x^\pi$, yang sering disebut sebagai *ukuran strategis*. Ekspektasi yang bersesuaian dinotasikan dengan $\mathbb{E}_x^\pi$.

Lebih jauh lagi, teorema Ionescu-Tulcea mengimplikasikan bahwa $\mathbb{P}_x^\pi$ membentuk sebuah probabilitas transisi reguler dari $(X, \mathcal{X})$ ke $(H_\infty, \mathcal{H}_\infty)$. Akibatnya, untuk setiap kebijakan π , fungsi nilai $v_n(x, \pi, \beta, f)$ dan $v(x, \pi, \beta)$ dapat diukur dalam x , dengan asumsi bahwa fungsi imbalan akhir f juga dapat diukur. Kami mencatat bahwa teorema Ionescu Tulcea digunakan dalam konstruksi kami, alih-alih teorema perluasan Kolmogorov yang lebih dikenal luas. Pilihan ini disengaja: sementara teorema Kolmogorov mengharuskan ruang keadaan memiliki struktur tertentu (yakni, harus berupa ruang Borel), teorema Ionescu Tulcea tidak memaksakan asumsi struktural seperti itu, menjadikannya dapat diterapkan secara lebih luas.

Dari sudut pandang intuitif, kebijakan acak lebih umum daripada kebijakan non-acak. Faktanya, setiap kebijakan non-acak dapat dipandang sebagai kasus khusus di dalam kelas kebijakan acak. Untuk mencegah kasus-kasus trivial atau terdegenerasi, sangat penting untuk memastikan bahwa MDP memiliki setidaknya satu kebijakan yang valid. Untuk tujuan ini, kami menetapkan dua asumsi yang ringan dan alami berikut:

- Setiap himpunan tunggal $\{a\}$, untuk $a \in A$, adalah elemen dari σ -field $\mathcal{A}$.
- Terdapat setidaknya satu pemetaan terukur $\phi: X \rightarrow A$ sedemikian sehingga $\phi(x) \in A(x)$ untuk semua $x \in X$.

Asumsi-asumsi ini selalu terpenuhi ketika model melibatkan ruang keadaan dan aksi diskrit. Kondisi kedua, khususnya, berlaku secara otomatis ketika ruang keadaan bersifat diskrit. Diberikan suatu ukuran ν dan suatu fungsi terukur f , kita mengadopsi konvensi notasi berikut:

$$\nu(f) = \int f(a) d\nu(a) = f(\nu), \quad (9.22)$$

di mana semua ekspresi merupakan representasi yang setara dari f ekspektasi di bawah ν . Sekarang, pertimbangkan *kebijakan stasioner acak* π , dan misalkan $\pi_x(\cdot) = \pi(\cdot | x)$ menunjukkan distribusi probabilitas atas tindakan pada keadaan x . Kebijakan ini menginduksi rantai Markov dengan probabilitas transisi $p(dy | x, \pi_x)$. Dalam kasus di mana ruang keadaan X diskrit, dinamika transisi dijelaskan oleh matriks transisi $P(\pi)$ dengan elemen yang diberikan oleh $p_{xy}(\pi_x)$.

Singkatnya, MDP, bersama dengan strategi dan fungsi tujuan terkaitnya, dapat dirumuskan di bawah asumsi yang sangat umum. Namun, menganalisis MDP dengan ruang keadaan terukur yang sepenuhnya sembarang menimbulkan tantangan yang signifikan. Masalah utama pertama adalah bahwa fungsi nilai V mungkin gagal untuk diukur, bahkan dalam masalah keputusan satu langkah sederhana. Tantangan kedua muncul dalam upaya untuk membangun kebijakan Markov acak yang setara untuk kebijakan sembarang yang diberikan - langkah kunci dalam analisis MDP. Konstruksi ini melibatkan pendefinisian probabilitas transisi $\mathbb{P}_x^\pi(da_n | x_n)$, reguler, yang mungkin tidak ada untuk ruang keadaan dan tindakan umum (yaitu, non-Borel).



Kedua komplikasi ini dapat dihindari ketika ruang keadaan dapat dihitung, dan juga dapat diselesaikan jika ruang keadaan dan ruang aksi keduanya terukur Borel. Saat ini, belum ada justifikasi teoritis atau praktis yang kuat untuk mempelajari MDP yang didefinisikan di atas ruang terukur yang sepenuhnya sembarang. Sebaliknya, MDP dengan ruang keadaan Borel memiliki aplikasi yang mapan di bidang-bidang seperti statistik, kontrol sistem dengan pengamatan parsial, dan manajemen inventaris. Namun, kita tidak mengetahui aplikasi yang berarti yang akan memerlukan pemodelan MDP dengan ruang keadaan dengan kardinalitas lebih besar dari kontinum.

MDP Keadaan Diskrit

Dalam pengaturan ini, ruang keadaan X diasumsikan diskrit, sedangkan ruang aksi adalah ruang terukur sedemikian $(A, \mathcal{A})$ rupa sehingga setiap himpunan $\{a\}$ tunggal terukur. Himpunan aksi yang layak $A(x)$ untuk setiap keadaan $x \in X$ juga diasumsikan sebagai elemen dari $\mathcal{A}$. Dalam kondisi ini, fungsi hadiah $r(x, a)$ dan probabilitas transisi $p(y | x, a)$ secara otomatis terukur terhadap a . Semua konstruksi yang dibahas sebelumnya untuk MDP diskrit dan umum tetap berlaku dalam kasus ini, dengan σ -field $\mathcal{X}$ pada X dianggap sebagai himpunan kuasa, yaitu, kumpulan semua himpunan bagian dari X .

MDP Borel Klasik

Dua ruang terukur $(E_1, \mathcal{E}_1)$ dan $(E_2, \mathcal{E}_2)$ dikatakan isomorfik jika terdapat pemetaan bijektif (satu-satu dan onto) sedemikian $f: E_1 \rightarrow E_2$ sehingga f terukur terhadap $\mathcal{E}_1, \mathcal{E}_2$, dan inversnya f^{-1} juga terukur. *Ruang Polish* didefinisikan sebagai ruang metrik lengkap dan terpisah. Kecuali dinyatakan sebaliknya, kita mengasumsikan bahwa setiap ruang E metrik dilengkapi dengan σ -field Borel-nya $\mathcal{B}(E)$, yang merupakan σ -field terkecil yang berisi semua himpunan bagian terbuka dari E .

Yang penting, setiap himpunan bagian terukur $E' \subseteq E$ dari *ruang Polish* itu sendiri E adalah *ruang Polish* ketika dilengkapi dengan σ -field Borel $\mathcal{B}(E')$, yang didefinisikan sebagai irisan E' dengan himpunan bagian Borel dari E . Suatu ruang terukur disebut $(E, \mathcal{E})$ sebagai *ruang Borel* jika isomorfik dengan ruang Polish. Ruang Borel termasuk ke dalam salah satu dari tiga kategori: berhingga, terhitung, atau tak terhitung dengan kardinalitas kontinum. Lebih jauh lagi, setiap dua ruang Borel yang tak terhitung isomorfik satu sama lain dan terhadap himpunan Borel standar seperti $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ atau $([0,1], \mathcal{B}([0,1]))$.

Asumsi untuk MDP Borel

Untuk MDP yang didefinisikan pada *ruang Borel*, asumsi berikut dibuat:

- Ruang keadaan X dan ruang tindakan A adalah ruang Borel dengan medan- σ Borel masing-masing $\mathcal{X}$ dan $\mathcal{A}$.
- $\mathcal{A}$. Grafik pemetaan aksi,

$$\text{Gr}A(x) = \{(x, a) \in X \times A \mid a \in A(x)\},$$

adalah himpunan bagian terukur dari $X \times A$, dan terdapat setidaknya satu fungsi seleksi terukur sedemikian $\phi: X \rightarrow A$ sehingga $\phi(x) \in A(x)$ untuk semua $x \in X$.



- iii. Fungsi imbalan $r(x, a)$ terukur pada $X \times A$, dan *probabilitas transisi* $p(\cdot | x, a)$ membentuk *probabilitas transisi reguler* dari $X \times A$ ke X .

Asumsi (i) dan (iii) mencerminkan asumsi-asumsi yang biasanya digunakan dalam kerangka kerja MDP yang lebih umum. Keterukuran grafik pada (ii) menjamin bahwa setiap himpunan tindakan yang layak $A(x)$ dapat diukur. Keberadaan penyeleksi yang dapat diukur memastikan bahwa $A(x) \neq \emptyset$ untuk semua $x \in X$.

Patut dicatat bahwa bahkan ketika grafik $\text{Gr } A(x)$ adalah Borel dan masing-masing $A(x)$ tidak kosong, mungkin saja tidak terdapat fungsi seleksi terukur Borel. Oleh karena itu, bagian kedua dari asumsi (ii) sangatlah penting untuk memastikan eksistensi setidaknya satu kebijakan yang valid. Sebagaimana telah dibahas sebelumnya, salah satu tantangan utama dalam MDP Borel adalah fungsi nilai V bisa saja gagal menjadi terukur secara Borel, bahkan pada masalah keputusan satu langkah. Akan tetapi, berdasarkan asumsi (i)–(iii), fungsi nilai dijamin dapat *terukur secara universal* baik untuk pengaturan horizon berhingga maupun tak berhingga. Sifat ini memungkinkan pendefinisian dan penerapan operator-operator optimalitas. Sekarang kita memperkenalkan konsep himpunan dan fungsi yang terukur secara universal.

Misalkan $(E, \mathcal{E})$ adalah suatu ruang Borel. Diberikan suatu ukuran probabilitas p pada ruang ini, kita mendefinisikan *penyelesaian* dari $\mathcal{E}$ terhadap p , yang dinotasikan dengan $\mathcal{E}_p$. Medan- σ yang telah diselesaikan ini adalah medan- σ terkecil yang memuat semua himpunan dalam $\mathcal{E}$ serta semua subhimpunan $F \subseteq E$ di mana terdapat suatu himpunan $F' \in \mathcal{E}$ sedemikian sehingga $F \subseteq F'$ dan $p(F') = 0$. Sebagai contoh, perhatikan ruang Borel $([0,1], \mathcal{B}([0,1]))$ dengan ukuran Lebesgue m yang didefinisikan oleh $m([a, b]) = |b - a|$. Penyelesaian $\mathcal{E}_m$ dalam kasus ini dikenal sebagai *medan- σ Lebesgue*. Misalkan $\mathbf{P}(E)$ menyatakan koleksi semua ukuran probabilitas pada E . Medan- σ *universal*, yang dinotasikan dengan $\mathcal{U}(E)$, didefinisikan sebagai irisan dari semua medan- σ yang diselesaikan $\mathcal{E}_p$ atas semua $p \in \mathbf{P}(E)$:

$$\mathcal{U}(E) = \bigcap_{p \in \mathbf{P}(E)} \mathcal{E}_p.$$

Medan- σ universal ini juga disebut sebagai medan- σ dari *himpunan terukur secara universal*, dan elemen-elemennya disebut *subhimpunan terukur secara universal* dari E . Suatu fungsi $f: X \rightarrow \mathbb{R}$ dikatakan *terukur secara universal* jika fungsi tersebut dapat diukur terhadap medan- σ universal $\mathcal{U}(X)$ pada X dan medan- σ Borel $\mathcal{B}(\mathbb{R})$ pada $\mathbb{R}$. Secara alami, setiap himpunan Borel dan setiap fungsi terukur Borel juga terukur secara universal. Meskipun persamaan optimalitas dapat dirumuskan untuk MDP Borel, terdapat suatu komplikasi yang halus namun patut dicatat—terutama mengenai masalah estetika. Secara spesifik, kebijakan ϵ -optimal mungkin tidak selalu ada untuk $\epsilon > 0$ yang sangat kecil sembarang, bahkan dalam MDP Borel satu langkah yang sederhana dengan fungsi imbalan terbatas.

Masalah ini diilustrasikan oleh sebuah contoh dari David Blackwell, yang menyoroti bahwa meskipun fungsi nilai dalam model-model semacam itu dapat terukur secara universal, fungsi tersebut tidak harus terukur secara Borel. Sebaliknya, untuk setiap kebijakan yang



diberikan, ekspektasi imbalan satu langkah tetap merupakan fungsi terukur Borel dari keadaan awal. Namun demikian, dapat ditunjukkan bahwa untuk setiap ukuran probabilitas awal p pada ruang keadaan X dan untuk setiap $\epsilon > 0$, terdapat suatu kebijakan yang ϵ -optimal terhadap p secara hampir pasti. Kebijakan semacam itu disebut sebagai *optimal*-(p, ϵ).

MDP Borel yang Terukur Secara Universal

Jika kita memperluas kelas kebijakan yang dapat diterima untuk mencakup kebijakan yang terukur secara universal, maka kebijakan ϵ -optimal dijamin keberadaannya. Dalam kerangka kerja yang lebih luas ini, gagasan mengenai optimalitas-(p, ϵ) menjadi tidak diperlukan. Namun, generalisasi ini mengharuskan kita untuk melonggarkan beberapa asumsi standar—khususnya asumsi (ii) dan (iii)—yang menjadi dasar ditetapkan hasil-hasil sebelumnya. Sebelum melanjutkan dengan definisi formal, kita akan meninjau secara singkat konsep *himpunan analitik*. Misalkan f adalah suatu pemetaan terukur dari ruang Borel $(E_1, \mathcal{E}_1)$ ke ruang Borel lainnya $(E, \mathcal{E})$. Untuk setiap himpunan Borel $F \in \mathcal{E}$, pracitra $f^{-1}(F)$ termasuk dalam $\mathcal{E}_1$ berdasarkan definisi. Namun, ada kemungkinan bahwa citra $f(E_1)$ bukan merupakan himpunan Borel dalam E . Suatu subhimpunan $F \subseteq E$ disebut *analitik* jika terdapat suatu ruang Borel $(E_1, \mathcal{E}_1)$ dan pemetaan terukur $f: E_1 \rightarrow E$ sedemikian sehingga $F = f(F_1)$ untuk suatu himpunan Borel $F_1 \in \mathcal{E}_1$. Karena kita selalu dapat memilih $E_1 = E$ dan $f(e) = e$, maka dapat disimpulkan bahwa setiap himpunan Borel adalah analitik. Lebih jauh lagi, setiap himpunan analitik terukur secara universal.

Selain kelas himpunan analitik, kita dapat mendefinisikan medan- σ dari *himpunan terukur secara analitik*, yang merupakan medan- σ terkecil yang memuat semua subhimpunan analitik dari suatu himpunan analitik yang diberikan. Tidak seperti medan- σ Borel dan universal—yang tertutup terhadap operasi komplemen dan masing-masing seluruhnya terdiri dari himpunan Borel atau himpunan terukur secara universal—medan- σ dari himpunan terukur secara analitik dapat mencakup himpunan-himpunan yang bukan analitik, karena komplemen dari suatu himpunan analitik mungkin saja tidak analitik. Terdapat beberapa definisi ekuivalen dari himpunan analitik. Sebagai contoh, dalam ruang Polish, himpunan analitik juga dapat dikarakterisasi sebagai citra kontinu atau proyeksi dari himpunan Borel. Jika $(E, \mathcal{E})$ dan $(E_1, \mathcal{E}_1)$ adalah ruang Borel (yakni, himpunan Borel yang dilengkapi dengan medan- σ Borel), maka suatu fungsi $f: E \rightarrow E_1$ dikatakan *terukur secara universal* (atau *terukur secara analitik*) jika, untuk setiap himpunan Borel $B \in \mathcal{E}_1$, pracitra $f^{-1}(B)$ termasuk dalam medan- σ dari subhimpunan yang terukur secara universal (berturut-turut, analitik) dari E .

Asumsi untuk MDP yang Terukur Secara Universal

Dalam konteks MDP yang terukur secara universal, asumsi berikut biasanya dibuat:

- Ruang keadaan $(X, \mathcal{X})$ dan ruang tindakan $(A, \mathcal{A})$ adalah ruang Borel.
- Grafik pemetaan aksi,

$$\text{Gr}A = \{(x, a) \in X \times A \mid a \in A(x)\},$$

adalah himpunan bagian analitik dari $X \times A$, dan setiap himpunan aksi yang $A(x)$ layak tidak kosong.



- c. Fungsi imbalan $r(x, a)$ *analitik atas* pada $X \times A$, yang berarti bahwa untuk setiap bilangan real c , himpunan $\{(x, a) \in X \times A \mid r(x, a) \geq c\}$ adalah analitik.
- d. Kernel transisi $p(\cdot \mid x, a)$ adalah probabilitas transisi reguler dari $(X \times A, \mathcal{X} \times \mathcal{A})$ ke $(X, \mathcal{X})$.

Asumsi (a) dan (d) sejalan dengan asumsi yang digunakan untuk MDP Borel standar. Menurut teorema Jankov–von Neumann, asumsi (b) memastikan eksistensi fungsi seleksi yang terukur secara analitik $\phi: X \rightarrow A$ sedemikian sehingga $\phi(x) \in A(x)$ untuk semua $x \in X$. Karena setiap fungsi yang terukur secara analitik juga terukur secara universal, hal ini menjamin bahwa setidaknya terdapat satu kebijakan yang dapat diterima. Asumsi (c) lebih umum daripada mensyaratkan fungsi imbalan $r(x, a)$ untuk terukur secara Borel. Namun, generalisasi ini tidaklah esensial dalam sebagian besar konteks praktis—hal ini utamanya dipertahankan dalam literatur karena hasil dan pembuktian teoretis yang sama berlaku baik untuk fungsi imbalan analitik atas maupun fungsi imbalan Borel.

Perbedaan utama antara MDP Borel dan MDP yang terukur secara universal terletak pada sifat dari kebijakan-kebijakannya. Dalam MDP yang terukur secara universal, semua kebijakan disyaratkan untuk dapat terukur secara universal:

- Kebijakan non-acak adalah fungsi-fungsi yang terukur secara universal $\phi_n: H_n \rightarrow A$ sedemikian sehingga $\phi_n(h_n) \in A(x_n)$ untuk setiap riwayat $h_n = x_0 a_0 \dots x_n$.
- Kebijakan Markov dan stasioner didefinisikan sebagai pemetaan yang terukur secara universal $\phi_n: X \rightarrow A$ (atau cukup ϕ untuk kebijakan stasioner) yang memenuhi $\phi_n(x) \in A(x)$ untuk semua $x \in X$.
- Kebijakan acak, Markov acak, dan stasioner acak didefinisikan melalui probabilitas transisi reguler, seperti pada kasus umum, tetapi ruang riwayat H_n dan ruang keadaan X sekarang dilengkapi dengan medan- σ dari subhimpunan yang terukur secara universal. Hal-hal ini memainkan peran yang sama dengan $\mathcal{E}_1$ dalam definisi standar dari probabilitas transisi reguler.

Asumsi (b) menjamin bahwa setidaknya terdapat satu kebijakan semacam itu. Formulasi alternatif dari MDP yang terukur secara universal juga tersedia. Sebagai contoh, kita dapat mendefinisikan kebijakan untuk dapat terukur secara analitik. Yang terpenting, semua definisi dan notasi yang dikembangkan untuk MDP diskrit dapat diperluas secara alami ke dalam pengaturan yang terukur secara universal.

9.2 PEMBELAJARAN PENGUATAN

Membangun konsep dasar MDP yang diperkenalkan pada bagian sebelumnya, kita sekarang mengalihkan perhatian ke *Reinforcement Learning* (RL)—sebuah kerangka kerja untuk mempelajari strategi pengambilan keputusan yang optimal melalui interaksi dengan lingkungan yang dimodelkan sebagai MDP. Ingat kembali bahwa MDP memperluas model Markov klasik dengan memasukkan tindakan. Di setiap status, sekumpulan tindakan yang mungkin dapat dijalankan, yang memengaruhi transisi ke status berikutnya. Secara formal, dinamika dari suatu MDP ditangkap oleh fungsi transisi $T(s, a, s')$, di mana:

- $s \in S$ adalah status saat ini,



- $a \in A$ adalah tindakan yang tersedia dalam status s , dan
- $s' \in S$ adalah kemungkinan status berikutnya yang dihasilkan dari penerapan tindakan a dalam status s .

Seperti pada model Markovian lainnya, *sifat Markov* berlaku: probabilitas transisi ke status berikutnya hanya bergantung pada status dan tindakan saat ini, serta tidak bergantung pada riwayat status atau tindakan sebelumnya. Hal ini secara formal dinyatakan sebagai:

$$P(S_t = s' \mid S_{t-1} = s, a_t = a) = T(s, a, s').$$

Selain transisi, setiap MDP dilengkapi dengan fungsi imbalan $R: S \rightarrow \mathbb{R}$, yang memberikan imbalan bernilai real untuk setiap status. Tujuan dari pengambil keputusan adalah dua kali lipat:

- a. Memaksimalkan imbalan segera $R(s)$ saat memasuki status s , dan
- b. Mengoptimalkan imbalan kumulatif atau jangka panjang yang terakumulasi seiring waktu.

Untuk mencapai hal ini, kita mendefinisikan suatu kebijakan—sebuah fungsi $\Pi: S \rightarrow A$ yang memetakan setiap status $s \in S$ ke suatu tindakan $a \in A$. Kebijakan ini menentukan strategi pengambilan keputusan, yang memandu pemilihan tindakan berdasarkan status saat ini. RL berfokus pada bagaimana mempelajari kebijakan semacam itu, terutama ketika fungsi transisi T dan fungsi imbalan R tidak diketahui sepenuhnya sebelumnya, dengan cara berinteraksi dengan lingkungan dan memperbaiki keputusan seiring berjalannya waktu.

9.3 ITERASI NILAI

Dalam RL, hasil dari tindakan—seperti imbalan—dan konsekuensi jangka panjang dari mengikuti kebijakan tertentu sering kali tidak dapat diamati secara langsung. Oleh karena itu, strategi yang efektif harus menyertakan mekanisme yang memungkinkan kebijakan dievaluasi dan ditingkatkan di tengah ketidakpastian mengenai imbalan di masa depan. Mekanisme-mekanisme ini membentuk inti dari teknik-teknik RL. Pendekatan yang diadopsi secara luas untuk menyeimbangkan imbalan langsung dan masa depan adalah melalui penggunaan *faktor diskon*, yang dilambangkan dengan γ , di mana $0 < \gamma < 1$. Tingkat diskonto ini mengurangi dampak imbalan yang diterima jauh di masa depan, sehingga memberikan prioritas lebih tinggi pada hasil yang lebih segera. Total imbalan yang diharapkan kemudian direpresentasikan sebagai jumlah terdiskonto:

$$\sum_{t=0}^{\infty} \gamma^t r_t$$

di mana r_t adalah imbalan yang diterima pada langkah waktu t , dan $t = 0$ berhubungan dengan waktu saat ini. Seiring bertambahnya t , suku γ^t menurun, yang mengurangi pengaruh imbalan tertunda terhadap tujuan keseluruhan.



Misalkan $V^\Pi(s)$ menunjukkan fungsi nilai di bawah kebijakan Π yang diberikan. Fungsi ini menetapkan untuk setiap *keadaan* $s \in S$ imbalan kumulatif yang diharapkan ketika memulai di keadaan s dan mengikuti kebijakan Π setelahnya. Secara formal, hal ini dinyatakan sebagai:

$$V^\Pi(s) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s, \Pi \right]$$

di mana $a = \Pi(s)$ adalah tindakan yang dipilih oleh kebijakan Π dalam keadaan s . Tujuan kita adalah untuk menentukan kebijakan optimal Π^* yang memaksimalkan imbalan yang diharapkan. Mengamati persamaan di atas, kita mencatat bahwa imbalan segera $R(s)$ tidak bergantung pada kebijakan yang dipilih—imbalan tersebut bersifat intrinsik terhadap keadaan s itu sendiri. Konsekuensinya, untuk memaksimalkan $V^\Pi(s)$, kita harus memilih tindakan a yang memaksimalkan nilai masa depan yang diharapkan:

$$V^{\Pi^*}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} T(s, a, s') V^{\Pi^*}(s').$$

Ekspresi ini sangat penting dalam menentukan *kebijakan optimal* dan membentuk dasar dari algoritma *iterasi nilai*. *Iterasi nilai* dimulai dengan tebakan awal untuk fungsi nilai dan memperbaikinya secara iteratif dengan memperbarui estimasi nilai hingga konvergensi tercapai dalam toleransi yang ditentukan.

Algoritma 1: Iterasi Nilai

Iterasi nilai memperbarui perkiraan nilai secara berulang. Setiap pembaruan membuat model sedikit lebih dekat pada keputusan terbaik menurut imbalan jangka panjang.

1. Inisialisasi fungsi nilai $V(s)$ secara sembarang untuk semua $s \in S$.
2. $V(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} T(s, a, s') V(s')$
3. $\Pi(s) \leftarrow \arg \max_{a \in A(s)} Q(s, a)$ konvergensi

Pada setiap iterasi, fungsi nilai $V(s)$ diperbarui dengan memilih tindakan $a \in A(s)$ yang menghasilkan pengembalian yang diharapkan tertinggi, berdasarkan perkiraan nilai masa depan saat ini. Kebijakan $\Pi^*(s)$ kemudian diperbarui untuk mencerminkan tindakan yang memaksimalkan nilai ini. Fungsi $Q(s, a)$ tersebut mewakili nilai yang diharapkan dari mengambil tindakan a dalam keadaan S , dan digunakan untuk memandu kebijakan menuju optimalitas.

9.4 PEMBELAJARAN-Q

Metode iterasi nilai yang dijelaskan sebelumnya mengasumsikan pengetahuan lengkap tentang fungsi transisi $T(s, a, s')$. Namun, dalam banyak skenario praktis, probabilitas transisi tidak diketahui sebelumnya. Dalam kasus seperti itu, kita dapat menggunakan *pembelajaran-Q*, sebuah algoritma RL bebas model, untuk mempelajari nilai tindakan secara langsung



melalui pengalaman. Pembelajaran-Q memperkirakan *fungsi nilai-tindakan* $Q(s, a)$, yang mewakili imbalan kumulatif yang diharapkan dari mengambil tindakan a pada keadaan s , dan selanjutnya mengikuti kebijakan yang optimal. Bentuk dasar dari aturan pembaruan ini meniru struktur persamaan optimalitas Bellman:

$$Q(s, a) = R(s) + \gamma \max_{a'} Q_{\text{old}}(s, a').$$

Formulasi ini secara agresif menimpa nilai-nilai sebelumnya, yang dapat menyebabkan ketidakstabilan dalam pembelajaran. Versi yang lebih kuat dan umum digunakan menggunakan rata-rata tertimbang untuk memperbarui nilai-nilai secara bertahap:

$$Q(s, a) \leftarrow \eta Q_{\text{old}}(s, a) + (1 - \eta) \left[R(s) + \gamma \max_{a'} Q_{\text{old}}(s', a') \right],$$

di mana $\eta \in (0,1)$ adalah tingkat pembelajaran, yang menentukan seberapa besar pengaruh informasi baru terhadap nilai- Q yang ada. Proses ini menghasilkan algoritma iteratif yang mirip dengan iterasi nilai, dengan satu perbedaan utama: tindakan dipilih oleh fungsi eksplorasi $f(s)$, yang biasanya mengikuti kebijakan $\Pi(s)$ sebagian besar waktu, tetapi sesekali memilih tindakan acak untuk mendorong eksplorasi dan mengurangi bias pengambilan sampel. Kode semu untuk algoritma pembelajaran-Q ditunjukkan di bawah ini.

9.5 PEMBELAJARAN TD

Keterbatasan yang mencolok dari iterasi nilai adalah lambatnya propagasi sinyal imbalan, terutama dari keadaan terminal kembali ke keadaan awal. Sebagai contoh, dalam sebuah MDP yang mewakili navigasi labirin, imbalan yang signifikan mungkin hanya diterima setelah mencapai keadaan tujuan di akhir urutan. Namun, mungkin diperlukan banyak iterasi agar pengaruh imbalan tersebut dapat mencapai keadaan awal. *Pembelajaran TD* mengatasi masalah ini dengan memperbarui estimasi fungsi nilai secara bertahap, menggunakan nilai estimasi dari keadaan berikutnya daripada menunggu satu episode penuh atau konvergensi jangka panjang. Aturan pembaruan TD untuk fungsi nilai terparameterisasi $V(s; w)$ (di mana w adalah vektor bobot) diberikan oleh:

$$\Delta w = \eta (R(s_t) + \gamma V(s_{t+1}) - V(s_t)) \sum_{k=1}^t \lambda^{t-k} \frac{\partial V_k}{\partial w},$$

di mana:

- η adalah tingkat pembelajaran,
- γ adalah faktor diskon,
- λ adalah parameter peluruhan jejak $0 < \lambda < 1$, dan
- V_k menunjukkan nilai estimasi pada langkah k .



Algoritma di bawah ini menguraikan proses pembelajaran TD:

Algoritma 3: Pembelajaran TD

Pembelajaran TD memperbaiki prediksi sedikit demi sedikit setelah melihat satu transisi baru. Metode ini praktis karena tidak perlu menunggu seluruh episode selesai.

1. Inisialisasi semua bobot w secara acak
2. Pilih keadaan saat ini s_t
3. Pilih tindakan $a_t = \Pi(s_t)$ dan amati keadaan baru s_{t+1}
4. Hitung $V(s_{t+1})$ menggunakan bobot saat ini **Jika** imbalan aktual $R(s_t)$ tersedia
5. Gunakan hal tersebut untuk memperbarui estimasi imbal hasil
6. Perbarui bobot:
- 7.

$$w \leftarrow w + \eta(R(s_t) + \gamma V(s_{t+1}) - V(s_t)) \sum_{k=1}^t \lambda^{t-k} \frac{\partial V_k}{\partial w}$$

kebijakan Π konvergen

9.6 LATIHAN

1. **Memahami komponen MDP:** Pertimbangkan sebuah MDP dengan karakteristik berikut:
 - Ruang keadaan $S = \{s_1, s_2\}$,
 - Ruang tindakan $A = \{a_1, a_2\}$,
 - Probabilitas transisi: $T(s_1, a_1, s_2) = 1, T(s_2, a_2, s_1) = 1$,
 - Fungsi imbalan: $R(s_1) = 2, R(s_2) = 5$,
 - Faktor diskon $\gamma = 0.9$.
 - a. Tuliskan persamaan Bellman untuk setiap keadaan di bawah kebijakan tetap $\Pi(s_1) = a_1, \Pi(s_2) = a_2$.
 - b. Hitung fungsi nilai $V^\Pi(s_1)$ dan $V^\Pi(s_2)$.
2. **Mengimplementasikan iterasi nilai:** Diberikan MDP 3-keadaan dengan imbalan seragam $R(s) = 1$ untuk semua $s \in \{s_1, s_2, s_3\}$ dan probabilitas transisi didefinisikan sedemikian rupa sehingga setiap tindakan mengarah secara seragam ke semua keadaan:

$$T(s, a, s') = \frac{1}{3}, \quad \forall s, a, s',$$

inisialisasi $V(s) = 0$ dan lakukan 2 iterasi dari iterasi nilai dengan $\gamma = 0.9$. Tunjukkan semua langkah perantara.

3. **Aturan Pembaruan-Q:** Misalkan nilai-Q saat ini untuk (s, a) adalah $Q(s, a) = 4$. Agen menerima imbalan $R(s) = 3$, berpindah ke keadaan s' di mana $\max_{a'} Q(s', a') = 6$, dengan $\gamma = 0.8$ dan tingkat pembelajaran $\eta = 0.5$.
 - a. Hitung nilai-Q yang diperbarui $Q(s, a)$ setelah satu langkah pembelajaran-Q.



4. **Konseptual pembelajaran TD:** Jelaskan mengapa pembelajaran TD dianggap sebagai kombinasi dari metode Monte Carlo dan pemrograman dinamis. Berikan satu keuntungan menggunakan pembelajaran TD dibandingkan keduanya.
5. **Penurunan Kebijakan dari Nilai-Q:** Diberikan nilai-Q berikut:

$$Q(s_1, a_1) = 5, \quad Q(s_1, a_2) = 3, \quad Q(s_2, a_1) = 1, \quad Q(s_2, a_2) = 4,$$

turunkan kebijakan serakah $\Pi^*(s)$ untuk setiap keadaan.



BAB 10

ALGORITMA PREDIKSI

10.1 REGRESI

Bab ini membahas secara mendalam kumpulan algoritma komprehensif yang penting untuk pemodelan prediktif, sebuah landasan dalam pengembangan sistem cerdas di berbagai domain aplikasi. Algoritma-algoritma ini membentuk dasar untuk memungkinkan sistem mengantisipasi hasil di masa depan, menjadikannya sangat diperlukan di berbagai bidang mulai dari keuangan dan perawatan kesehatan hingga robotika dan pemrosesan bahasa alami. Kategori utama algoritma yang dibahas dalam bab ini meliputi: Dalam penggunaan nyata, model prediksi perlu diuji dengan data baru. Model yang terlalu cocok pada data pelatihan sering terlihat bagus di awal, tetapi mudah gagal ketika kondisi berubah.

- **Algoritma Regresi:** Algoritma ini dirancang untuk memprediksi nilai kontinu, seperti harga saham, suhu, atau kemungkinan pelanggan berhenti berlangganan.
- **Algoritma Klasifikasi:** Algoritma ini berfokus pada pengkategorian data ke dalam kelas atau label diskrit, seperti mengidentifikasi email *spam*, mendiagnosis penyakit, atau mengklasifikasikan gambar.
- **Algoritma Tanpa Pengawasan:** Algoritma ini mengungkap pola dan struktur tersembunyi dalam data tanpa pengetahuan sebelumnya tentang hasil yang diharapkan. Algoritma ini digunakan untuk tugas-tugas seperti segmentasi pelanggan, deteksi anomali, dan pengurangan dimensi.

Bagian-bagian selanjutnya dari bab ini akan membahas lebih dalam setiap kategori algoritma ini, memberikan gambaran rinci tentang prinsip, metodologi, dan aplikasi praktisnya. Regresi dipakai ketika kita ingin memprediksi nilai numerik. Fokusnya bukan hanya mencari garis atau kurva yang pas, tetapi juga memastikan model tetap masuk akal saat bertemu data baru.

Metode Kuadrat Terkecil dan Tetangga Terdekat

Metode kuadrat terkecil memilih model dengan jumlah kesalahan kuadrat paling kecil. Karena kesalahan besar diberi hukuman lebih berat, metode ini sensitif terhadap nilai ekstrem. Mengapa model linier sederhana menghasilkan batas keputusan linier dalam tugas klasifikasi? Untuk data di mana setiap kelas terdiri dari titik-titik yang diambil dari distribusi Gauss, mengapa model linier yang disebutkan sebelumnya tampak paling optimal untuk batas keputusan? Mengapa batas keputusan metode tetangga terdekat bersifat nonlinier, dan apakah ada skenario di mana batas keputusan tersebut dapat menjadi linier? Berapa jumlah parameter dalam metode k -tetangga terdekat? Dengan mempertimbangkan tujuan prediksi melalui penangkapan ranah hubungan $y = f(x)$, bagaimana perbandingan antara model linier dan metode k -tetangga terdekat?

Teori Prediksi

Teori prediksi mengingatkan bahwa model harus diuji pada data yang belum pernah dilihat. Nilai galat pelatihan yang kecil belum tentu berarti model siap dipakai. Makna statistik



dan matematis dari fungsi kerugian dan kesalahan prediksi yang diharapkan perlu dipertimbangkan. Pertimbangan kerugian kuadrat yang diberikan oleh:

$$\varepsilon(f) = \int [y - f(x)]^2 \Pr(dx, dy). \quad (10.1)$$

Kita menanyakan dalam keadaan apa peminimal dari ε ada di f , khususnya ketika

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \varepsilon(f),$$

keberadaannya, sifatnya, dan kelebihan serta kekurangan dari kerugian kuadrat. Selanjutnya, pendekatan $\hat{f}$ pada tetangga terdekat atau pemrediksi linier dalam bentuk $x^T \beta$ dengan β sebagai parameter juga perlu dibahas. Selain itu, kasus yang berkaitan dengan klasifikasi variabel kategorikal dipertimbangkan. Hal ini juga menimbulkan pertanyaan tentang keabsahan pertimbangan sebagai berikut:

$$\varepsilon_q(f) = \int |y - f(x)|^q \Pr(dx, dy) \quad \text{where } q \in (0, \infty).$$

Kutukan Dimensi

Kutukan dimensi muncul ketika jumlah fitur terlalu banyak dibanding jumlah data. Pada kondisi ini, data terlihat jarang dan model lebih mudah menangkap pola palsu. Apa itu "kutukan dimensi"? Apa penyebabnya? Dapatkah Anda membuktikannya $\frac{\pi^{p/2}}{\Gamma(\frac{p}{2}+1)2^p}$? Di mana p : dimensi, π : konstanta, 2^p : volume hiperkubus, $\Gamma(\frac{p}{2}+1)$ Fungsi Gamma. Bagaimana dimensi yang besar berperan tidak menguntungkan dalam model linier? Bagaimana dimensi yang besar berperan tidak menguntungkan dalam metode tetangga terdekat? Dapatkah Anda menjelaskan interaksi antara dimensi, distribusi probabilitas yang mendasari, skema pengambilan sampel, dan metode yang digunakan?

Pembelajaran sebagai Aproksimasi Fungsi

Mengapa "pembelajaran melalui contoh" dapat dianggap sebagai "aproksimasi fungsi"? Mengapa ada kebutuhan untuk aproksimasi? Mari kita tinjau kembali fungsi kerugian $\varepsilon_2(f) = \int [y - f(x)]^2 \Pr(dx, dy)$. Berapa banyak model yang mampu menyesuaikan sekumpulan titik data terbatas yang ditentukan? Konsep regularisasi dinyatakan sebagai $R_\lambda(f) = \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \int [f''(x)]^2 dx$. Apa tujuan dari regularisasi? Dapatkah Anda menjelaskan atribut f dari masing-masing kendali berikut: $\int [f''(x)]^2 dx$, $\int [f'(x)]^2 dx$, $\int [f(x)]^2 dx$? Misalkan $z_i = (y_i, f(x_i))$ untuk $i = 1, \dots, N$. Secara umum, regularisasi diberikan oleh $R_\lambda(f) = \theta(z_1, \dots, z_N) + \lambda \Psi(f)$. Mengapa penting untuk menyeimbangkan tarik-ulur bias-varians dengan benar? Selain itu, apakah Anda familier dengan pelembut (*mollifier*) dan, jika demikian, dapatkah Anda menjelaskan hubungan antara pelembut dan metode kernel? Parameter penghalusan mengatur seberapa lentur model mengikuti data.



Nilai yang terlalu kecil membuat kurva berderau, sedangkan nilai terlalu besar dapat menyembunyikan pola penting.

Rumus Kunci

Jika $\epsilon_2(f) = \int [y - f(x)]^2 \Pr(dx, dy)$ dan $\hat{f} = \arg \min_{f \in \mathcal{F}} \epsilon(f)$ ada, maka $\hat{f} = E[Y|X = x]$. Untuk k -tetangga terdekat, $\hat{f}$ didekati oleh $\text{Ave}(y_i | x_i \in N_k(x))$, dan untuk model linier, $\hat{f}$ didekati oleh $x^T \beta$, di mana $\beta = [E(X^T X)]^{-1} E(XY)$. Untuk klasifikasi, $\hat{f}$ diberikan oleh aturan $G(x) = \arg \min_{g \in \mathcal{G}} [1 - \Pr(g|X = x)]$, yang setara dengan $\hat{G}(x) = \mathcal{G}_k$ jika $\Pr(g_k|X = x) = \max_{g \in \mathcal{G}} \Pr(g|X = x)$. Dekomposisi bias-varians dapat dinyatakan sebagai $MSE(x_0) = E_{\mathcal{F}}[f(x_0) - \hat{y}_0]^2 = \text{Var}_{\mathcal{F}}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0)$. Untuk jarak median dari titik asal ke titik data terdekat di antara N titik independen dan terdistribusi identik dalam bola satuan berdimensi- p B_p , jaraknya diberikan oleh $d(p, N) = (1 - 2^{-1/N})^{1/p}$. Kesalahan prediksi untuk kuadrat terkecil linier diberikan oleh $\epsilon(x_0) = \sigma^2 + E_{\mathcal{F}} x_0^T (X^T X)^{-1} x_0 \sigma^2 + 0^2$. Untuk kernel Gauss, $K_{\lambda}(x_0, x) = \frac{1}{\lambda} \exp\left(-\frac{\|x - x_0\|^2}{2\lambda}\right)$ dan penaksir Nadaraya-Watson adalah $\hat{f}(x_0) = \frac{\sum_{i=1}^N K_{\lambda}(x_0, x_i) y_i}{\sum_{i=1}^N K_{\lambda}(x_0, x_i)}$. Jumlah fitur yang besar tidak selalu membuat model lebih pintar. Jika datanya terbatas, terlalu banyak fitur justru dapat membuat model menangkap hubungan kebetulan belaka.

Regresi Linier dan Kuadrat Terkecil

Misalkan $x = (1, x_1, x_2, \dots, x_p)$ adalah vektor p prediktor dan $\beta = (\beta_0, \beta_1, \beta_2, \dots, \beta_p)$ adalah vektor koefisien p yang tidak diketahui. Misalkan kita memiliki model

$$f(x) = \beta_0 + \sum_{j=1}^p x_j \beta_j = \langle x, \beta \rangle.$$

Selanjutnya, kita memiliki pengamatan (x_i, y_i) , $i = 1, 2, \dots, N$ untuk (x, y) sedemikian rupa sehingga $x_i = (1, x_{i1}, x_{i2}, \dots, x_{ip})$. Misalkan $\mathbf{X}$ adalah matriks desain yang dibentuk dari x_i dan $y = (y_1, y_2, \dots, y_N)$.

- Kapan $\mathbf{X}^T \mathbf{X}$ tidak dapat dibalik? Saat memperoleh estimasi kuadrat terkecil dari β , bagaimana jika $\mathbf{X}^T \mathbf{X}$ tidak dapat dibalik? Apa yang akan Anda lakukan dalam kasus ini?
- Apa pemahaman Anda tentang istilah "derajat kebebasan?"
- Berapa derajat kebebasan untuk

$$\hat{\sigma}^2 = \frac{1}{N - p - 1} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

di mana nilai yang $\hat{y}_i$ sesuai untuk y_i . Misalkan tr adalah operator jejak untuk matriks. Tahukah Anda bahwa

$$\text{tr}(I - X(X^T X)^{-1} X^T) = N - (p + 1), \quad (10.2)$$



ketika $N \geq p + 1$ dan $\text{rank}(\mathbf{X}) = p + 1$? sering disebut $H = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ sebagai “matriks topi.”

d. Misalkan estimasi kuadrat terkecil dari β ada secara unik, yaitu,

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

Metode kuadrat terkecil dapat diterapkan sebagai algoritma, terlepas dari apakah ada unsur acak yang terkait dengan pengamatan (x_i, y_i) . Jika nilai Y_i yang diberikan pada x_i mengikuti distribusi probabilitas tertentu, maka perlu untuk menilai akurasi dan stabilitas $\hat{\beta}$. Bagaimana menilai signifikansi $\hat{\beta}$ ketika Y_i adalah saling bebas dan terdistribusi identik Normal $(0, \sigma^2)$? Model linier sederhana dengan respons skalar dapat diperluas menjadi model linier dengan respons vektor, yaitu,

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E},$$

yang mana estimasi kuadrat terkecil $\hat{\mathbf{B}}$ dari $\mathbf{B}$, jika ada secara unik, diberikan oleh

$$\hat{\mathbf{B}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}. \quad (10.3)$$

$\hat{\mathbf{B}}$ diperoleh dari meminimalkan

$$g(\mathbf{B}) = \text{tr}(\mathbf{Y} - \mathbf{XB})^T (\mathbf{Y} - \mathbf{XB}).$$

Perhatikan bahwa penurunan $\hat{\mathbf{B}}$ menggunakan sifat-sifat operator jejak tr dan turunan matriks. Berikut beberapa teknik yang berguna untuk menangani turunan matriks. Misalkan $\mathbf{A} = (a_{ij})$ adalah matriks dan $g(\mathbf{A})$ adalah fungsi skalar dari $\mathbf{A}$, maka turunan parsialnya adalah:

$$\frac{\partial}{\partial a_{ij}} g(\mathbf{A}) = \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} [g(\mathbf{A} + \epsilon \mathbf{P}_{ij}) - g(\mathbf{A})],$$

di mana $\mathbf{P}_{ij}$ adalah matriks (i, j) yang entri tak nolnya hanya berupa entri dan bernilai 1. Pada dasarnya, $\{\mathbf{P}_{ij}\}$ adalah basis untuk ruang $\mathcal{A}$ matriks yang dimensinya sama dengan dimensi $\mathbf{A}$. Perhatikan bahwa $\text{tr}(\mathbf{A}^T \mathbf{C})$ untuk $\mathbf{A}, \mathbf{C} \in \mathcal{A}$ adalah hasil perkalian dalam pada ruang $\mathcal{A}$ (linear (nonkomutatif terhadap perkalian matriks)).

Pemilihan Variabel

Pemilihan variabel membantu model tetap ringkas. Fitur yang tidak relevan dapat membuat model lambat, sulit dibaca, dan lebih mudah salah saat digunakan. Pertimbangkan model linear yang diberikan pada Persamaan (10.4), yaitu,



$$f(x) = \beta_0 + \sum_{j=1}^p x_j \beta_j = \langle x, \beta \rangle. \quad (10.4)$$

Pemilihan variabel diperlukan untuk meningkatkan akurasi prediksi dan interpretasi.

Pemilihan Subset Terbaik dan Pemilihan Bertahap Maju dan Mundur

Untuk setiap r , misalkan A_r adalah himpunan semua subset dengan kardinalitas r dari $\mathbb{N}_p = \{1, 2, \dots, p\}$. Maka terdapat model-model $\binom{p}{r} \mathcal{M}_r$ berbentuk Persamaan (10.4), masing-masing memiliki r prediktor, dan kita dapat menemukan model terbaik f_r^* dari setiap kelas $\mathcal{M}_r$. Saat mempertimbangkan pemilihan variabel atau model, penting untuk menganalisis kriteria yang memandu proses pemilihan. Tantangan utama dalam pemilihan subset muncul dari kompleksitas penentuan subset prediktor yang optimal. Kesalahan kuadrat rata-rata dari koefisien yang diestimasi diamati mendekati untuk metode pemilihan subset terbaik, bertahap maju, dan bertahap mundur. Buku teks menjelaskan hal ini dengan menyatakan, “Kinerja mereka sangat mirip, seperti yang sering terjadi.” Namun, ada situasi di mana kinerja mereka dapat berbeda secara signifikan, dan akan sangat membantu untuk memberikan contoh kasus tersebut. Lebih lanjut, pemilihan bertahap mundur tidak dapat diterapkan ketika $N < p$, artinya ukuran sampel lebih kecil daripada jumlah prediktor. Di antara pemilihan subset terbaik, pemilihan bertahap maju, dan pemilihan bertahap mundur, penting untuk mengidentifikasi metode mana yang menghasilkan urutan model bersarang. Terakhir, pertimbangkan metode pemilihan variabel yang menyertakan prediktor jika prediktor tersebut paling berkorelasi dengan residual saat ini di antara semua prediktor yang tersisa. Bagaimana pendapat Anda tentang efektivitas metode tersebut? Pertimbangkan modelnya

$$y_{N \times 1} = X_{N \times p} \beta_{p \times 1} + \epsilon_{N \times 1}.$$

Norma- q dari vektor $\beta = (\beta_1, \dots, \beta_p)$ untuk $q \in [0, \infty]$ didefinisikan sebagai:

$$\|\beta\|_q = \left(\sum_{j=1}^p |\beta_j|^q \right)^{1/q}. \quad (10.5)$$

Penting untuk dicatat bahwa $\|\beta\|_0$ bersesuaian dengan jumlah entri bukan nol dari β , sementara $\|\beta\|_\infty$ merepresentasikan nilai maksimum di antara $\{|\beta_j|\}_{j=1}^p$. Regresi ridge, yang juga dikenal sebagai penalti- L_2 pada β , meminimalkan fungsi objektif berikut:

$$g(\beta; \lambda, 2) = \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 = \|y\|_2^2 - 2\beta^T X^T y + \beta^T (X^T X + \lambda I) \beta, \quad (10.6)$$

di mana $\lambda \geq 0$ L_1 - β , adalah parameter regularisasi. LASSO, atau L_1 -penalti pada β , didefinisikan oleh fungsi tujuan:



$$g(\beta; \lambda, 1) = \| \mathbf{y} - \mathbf{X}\beta \|_2^2 + \lambda \| \beta \|_1, \quad (10.7)$$

di mana $\lambda \geq 0$. L_0 - β , adalah parameter regularisasi. LASSO, atau L_0 -penalti pada β , didefinisikan oleh fungsi tujuan:

$$g(\beta; \lambda, 0) = \| \mathbf{y} - \mathbf{X}\beta \|_2^2 + \lambda \| \beta \|_0.$$

Namun, penting untuk dicatat bahwa $g(\beta; \lambda, 0)$ tidak cembung di β . Jika kita ingin meminimalkan $g(\beta; \lambda, 2)$ atau $g(\beta; \lambda, 1)$, keduanya adalah masalah optimasi cembung di β . Namun, $g(\beta; \lambda, 1)$ β , tidak mulus di , karena memiliki titik puncak pada $\beta_j = 0$ untuk setiap j . Metode kuadrat terkecil pada dasarnya bersifat geometris, dan untuk regresi Ridge solusinya $\hat{\beta}^{\text{ridge}}$ memiliki fitur geometris yang menarik. Ingat

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}. \quad (10.8)$$

Misalkan $\mathbf{X}$ memiliki dekomposisi nilai singular (SVD)

$$\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T,$$

di mana kolom-kolom dari $\mathbf{U}_{N \times p}$ saling ortogonal satu sama lain, kolom-kolom dari $\mathbf{V}_{p \times p}$ saling ortogonal satu sama lain, dan $\mathbf{D}_{p \times p}$ adalah matriks diagonal. Kemudian,

$$\mathbf{X} \hat{\beta}^{\text{ridge}} = \sum_{j=1}^p \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^T \mathbf{y}. \quad (10.9)$$

Misalkan $\hat{\beta}^{\text{ls}} = (\hat{\beta}_1, \dots, \hat{\beta}_p)$ sebagai estimasi kuadrat terkecil. Ketika $\mathbf{X}^T \mathbf{X} = \mathbf{I}$,

- untuk *Ridge*,

$$\hat{\beta}_j^{\text{ridge}} = \frac{1}{1 + \lambda} \hat{\beta}_j^{\text{ls}}. \quad (10.10)$$

Umumnya,

$$\begin{aligned} \hat{\beta}^{\text{ridge}} &= (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \\ &= (\mathbf{I} + \lambda (\mathbf{X}^T \mathbf{X})^{-1})^{-1} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= (\mathbf{I} + \lambda (\mathbf{X}^T \mathbf{X})^{-1})^{-1} \hat{\beta}^{\text{ls}} \end{aligned}$$

- Untuk *LASSO*,

$$\hat{\beta}_j^{\text{lasso}} = \text{sgn}(\hat{\beta}_j^{\text{ls}}) (|\hat{\beta}_j^{\text{ls}}| - \lambda)_+. \quad (10.11)$$



di mana fungsi sgn tanda bertindak per entri, fungsi nilai $|\cdot|$ absolut juga per entri dan $(x)_+$ merupakan bagian positif dari x .

Untuk pemilihan subset model berukuran M ,

$$\hat{\beta}_j^{\text{subset}} = \hat{\beta}_j^{\text{ls}} \mathbf{1}_{\{|\hat{\beta}_j^{\text{ls}}| \geq \gamma_M\}}, \quad (10.12)$$

di mana $\gamma_M > 0$ adalah konstanta untuk semua model berukuran M dan $\mathbf{1}_A$ adalah indikator dari suatu himpunan A .

Deviasi Absolut yang Dipangkas Halus

Estimator yang dihasilkan memiliki beberapa sifat yang diinginkan. Pertama, estimator ini hampir tidak bias ketika parameter sebenarnya yang tidak diketahui berukuran besar, yang membantu menghindari bias pemodelan yang tidak perlu. Bagian ini membantu pembaca melihat perbedaan antara model yang stabil dan model yang terlalu peka terhadap data pelatihan. Validasi menjadi alat penting untuk memeriksa hal tersebut. Kedua, estimator ini menunjukkan kelangkaan, karena bertindak sebagai aturan ambang batas yang secara otomatis menetapkan koefisien estimasi kecil menjadi nol, sehingga mengurangi kompleksitas model. Terakhir, estimator ini kontinu terhadap data, yang membantu mencegah ketidakstabilan dalam prediksi model. Deviasi absolut yang dipangkas halus (SCAD) dirancang untuk mengatasi bias *LASSO* ketika koefisien sebenarnya memiliki besaran yang besar. Ketika $\mathbf{X}^T \mathbf{X} = \mathbf{I}$, estimasi SCAD diberikan oleh:

$$\hat{\beta}_j^{\text{scad}} = \begin{cases} \text{sgn}(\hat{\beta}_j^{\text{ls}})(|\hat{\beta}_j^{\text{ls}}| - \lambda)_+, & \text{if } |\hat{\beta}_j^{\text{ls}}| \leq 2\lambda, \\ \frac{(a-1)\hat{\beta}_j^{\text{ls}} - \text{sgn}(\hat{\beta}_j^{\text{ls}})a\lambda}{a-2}, & \text{if } 2\lambda < |\hat{\beta}_j^{\text{ls}}| \leq a\lambda, \\ z, & \text{if } |z| > a\lambda, \end{cases} \quad (10.13)$$

di mana notasi tersebut mewakili fungsi tanda, adalah parameter regularisasi, dan adalah parameter penyetelan yang mengontrol perilaku ambang batas SCAD.

Konsistensi dan Properti Oracle

Estimator $\hat{\theta}_n$ parameter θ yang tidak diketahui berdasarkan sampel berukuran dianggap n konsisten jika, untuk konstanta tertentu $\epsilon > 0$, probabilitas bahwa perbedaan absolut antara $\hat{\theta}_n$ dan θ melebihi ϵ mendekati nol ketika ukuran sampel cenderung n tak hingga. Secara formal, ini dinyatakan sebagai:

$$\lim_{n \rightarrow \infty} \mathbb{P}(|\hat{\theta}_n - \theta| > \epsilon) = 0.$$

Selain itu, estimator dikatakan konsisten secara kuat jika probabilitas bahwa selisih absolut antara $\hat{\theta}_n$ dan θ konvergen ke nol saat $n \rightarrow \infty$ adalah sama dengan satu:



$$\mathbb{P} \left(\lim_{n \rightarrow \infty} |\hat{\theta}_n - \theta| = 0 \right) = 1.$$

Sekarang pertimbangkan kasus di mana parameter θ dapat diuraikan menjadi dua subvektor, yang dilambangkan sebagai $\theta = (\theta_1, \theta_0)$, di mana $\theta_0 = 0$ dan tidak ada entri dari θ_1 yang bernilai nol. Dalam konteks ini, estimator $\hat{\theta}_n$ dikatakan memenuhi "sifat oracle" jika ia juga dapat diuraikan menjadi dua subvektor, $\hat{\theta}_n = (\hat{\theta}_{1,n}, \hat{\theta}_{0,n})$, sedemikian sehingga $\hat{\theta}_{0,n} = 0$ dan

$$\frac{\hat{\theta}_{1,n} - \theta_1 - a_n}{b_n} \rightsquigarrow \text{Normal}(0, \Sigma),$$

di mana $\rightsquigarrow$ menunjukkan konvergensi dalam distribusi (atau konvergensi lemah), dan $\{a_n\}_{n \geq 1}$ dan $\{b_n\}_{n \geq 1}$ adalah dua urutan.

Memilih Sekelompok Variabel

Misalkan p prediktor dibagi menjadi beberapa L kelompok, dengan p_l jumlah dalam kelompok l . Biarkan $\mathbf{X}_l$ mewakili prediktor yang sesuai dengan kelompok l , dengan vektor koefisien yang sesuai β_l . *Lasso* berkelompok meminimalkan kriteria cembung:

$$\min_{\beta \in \mathbb{R}^p} \left(\left\| \mathbf{y} - \beta_0 \mathbf{1} - \sum_{l=1}^L \mathbf{X}_l \beta_l \right\|_2^2 + \lambda \sum_{l=1}^L \sqrt{p_l} \|\beta_l\|_2 \right). \quad (10.14)$$

Kuadrat Terkecil, Kemungkinan yang Dikenai Penalti, dan Inferensi Bayesian

Metode kuadrat terkecil memilih model dengan jumlah kesalahan kuadrat paling kecil. Karena kesalahan besar diberi hukuman lebih berat, metode ini sensitif terhadap nilai ekstrem. Pertimbangkan modelnya

$$\mathbf{y}_{N \times 1} = \mathbf{X}_{N \times p} \boldsymbol{\beta}_{p \times 1} + \boldsymbol{\varepsilon}_{N \times 1},$$

di mana $\boldsymbol{\varepsilon} \sim \mathcal{N}_N(0, I)$. Maka kemungkinan logaritmiknya adalah

$$\log L(\beta) \propto \|\mathbf{y} - \mathbf{X}\beta\|_2^2.$$

Jika $\beta \sim \text{Normal}\left(0, \frac{\lambda}{2} \mathbf{I}\right)$, maka kemungkinan logaritmik gabungan adalah

$$\log L_2(\mathbf{y}, \beta) \propto \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_2^2.$$

Jika $\beta_i \sim \text{Laplace}(0, \lambda)$ dengan kepadatan, maka kemungkinan logaritmik gabungan adalah



$$\log L_1(\mathbf{y}, \beta) \propto \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1$$

Misalkan fungsi kepadatan/massa probabilitas bersyarat dari $\mathbf{y}$ diberikan β adalah $p(\mathbf{y} | \beta)$ dan bahwa untuk β ketika $p(\beta)$, maka distribusi posterior dari β yang $\mathbf{y}$ diberikan didefinisikan sebagai:

$$p(\beta | \mathbf{y}) = \frac{p(\mathbf{y} | \beta)p(\beta)}{\int p(\mathbf{y} | \beta)p(\beta) d\beta} = \frac{p(\mathbf{y} | \beta)p(\beta)}{p(\mathbf{y})}.$$

Kesalahan kuadrat rata-rata (MSE) didefinisikan sebagai:

$$\begin{aligned} \mathbb{E}[\|\hat{\beta} - \beta\|^2] &= \iint \|\hat{\beta} - \beta\|^2 p(\mathbf{y} | \beta)p(\beta) d\mathbf{y} d\beta \\ &= \iint \|\hat{\beta} - \beta\|^2 dF(\mathbf{y}, \beta) \end{aligned} \quad (10.15)$$

Rata-rata posterior:

$$\hat{\beta}_0 = \mathbb{E}[\beta | \mathbf{y}] = \int \beta p(\beta | \mathbf{y}) d\beta \quad (10.16)$$

meminimalkan MSE dan disebut estimasi Bayes (untuk MSE). Derivasi Persamaan (10.16) menggunakan apa yang disebut "sifat menara dari ekspektasi bersyarat."

10.2 KLASIFIKASI UBAH DATA

Klasifikasi mengubah data menjadi label. Dalam praktik, kesalahan pada satu label bisa lebih mahal daripada label lain, sehingga ambang keputusan perlu dipilih dengan hati-hati. Pada bagian ini, kita fokus pada metode linier untuk klasifikasi. Mari kita pertimbangkan masalah klasifikasi yang melibatkan dua kelas, yang dilambangkan sebagai $\{1, 2\}$. Salah satu pendekatan yang mungkin untuk memodelkan masalah ini didasarkan pada probabilitas posterior. Secara khusus, model tersebut dapat dinyatakan sebagai:

$$\log \frac{\Pr(G = 1 | X = x)}{\Pr(G = 2 | X = x)} = \beta_0 + \beta^T x, \quad (10.17)$$

atau dengan kata lain,

$$\log \frac{p(x)}{1 - p(x)} = \beta_0 + \beta^T x,$$

di mana $p(x) = \Pr(G = 1 | X = x)$. Model ini dicirikan oleh batas keputusan linier dan merupakan bagian dari kelas metode yang dikenal sebagai *analisis diskriminan linier* (LDA).



Masalah dengan Pendekatan Regresi Linier

Misalkan $\mathcal{G}$ terdapat K kelas. Untuk setiap observasi Y , vektor $\mathbf{Y} = (Y_1, \dots, Y_K)$ indikator diinduksi, dan untuk N observasi, ini menghasilkan matriks indikator $\mathbf{Y}$. Jika regresi linier dengan intersep dipasang menggunakan metode kuadrat terkecil, nilai yang dipasang diberikan oleh $\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$, sedangkan matriks koefisien regresi direpresentasikan sebagai $\hat{\mathbf{B}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$. Untuk input x yang diberikan, fungsi yang diprediksi dinyatakan sebagai $\hat{f}(x)^T = (\hat{f}_1(x), \dots, \hat{f}_K(x))^T = (1, x^T) \hat{\mathbf{B}}$. dan klasifikasi dilakukan berdasarkan:

$$\hat{G}(x) = \arg \max_{k \in \mathcal{G}} \hat{f}_k(x).$$

Alternatif lain, seseorang dapat meminimalkan fungsi objektif $\min_{\mathbf{B}} \sum_{i=1}^N \left\| y_i - ((1, x_i^T) \mathbf{B})^T \right\|^2$, di mana $y_i = t_k$ jika $g_i = k$, dan melakukan klasifikasi berdasarkan:

$$\hat{G}(x) = \arg \max_{k \in \mathcal{G}} \left\| \hat{f}(x) - t_k \right\|.$$

Analisis Diskriminan Linier

Model yang digunakan secara luas untuk masalah klasifikasi adalah model campuran. Dalam kerangka kerja ini, misalkan $\pi_k = \Pr(G = k)$ mewakili probabilitas prior untuk kelas k , dan misalkan $f_k(x)$ menyatakan densitas kondisional dari X jika diketahui $G = k$, yaitu, $f_k(x) = \Pr(X \approx x \mid G = k)$. Probabilitas posterior kemudian dapat dinyatakan sebagai:

$$\Pr(G = k \mid X = x) = \frac{f_k(x) \pi_k}{\sum_{l=1}^K f_l(x) \pi_l}, \quad (10.18)$$

di mana K adalah jumlah total kelas. Dengan asumsi distribusi bersyarat Gaussian, fungsi densitasnya $f_k(x)$ berbentuk:

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} \exp \left(-\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right).$$

ketika $\sigma_k = \Sigma$ (yaitu, matriks kovarians bersama antar kelas), model tersebut sesuai dengan LDA. Dalam kasus ini, probabilitas log-posterior disederhanakan menjadi:

$$\log \Pr(G = k \mid X = x) \propto x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k = \delta_k(x), \quad (10.19)$$

di mana fungsi $\delta_k(x)$ diskriminan bersifat linier terhadap x . Untuk masalah klasifikasi biner, aturan keputusan yang tersirat dalam Persamaan (10.19) dapat disederhanakan lebih lanjut, seperti yang dibahas dalam teks.



Catatan 10.1

Terdapat banyak literatur tentang model campuran untuk masalah klasifikasi. Metode analisis diskriminan yang disajikan dalam teks biasanya diimplementasikan menggunakan estimasi kemungkinan maksimum (MLE).

Ketika matriks kovariansi σ_k tidak sama di seluruh kelas, model tersebut sesuai dengan analisis diskriminan kuadratik (QDA). Dalam hal ini, fungsi diskriminan diberikan oleh:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \log\pi_k.$$

Batas keputusan antara kelas k dan l didefinisikan oleh himpunan titik x , di mana

$$\delta_k(x) = \delta_l(x).$$

Salah satu tantangan dalam mengimplementasikan QDA adalah estimasi semua matriks kovarians Σ_k , terutama ketika dimensinya p bernilai tinggi.

LDA Peringkat Tereduksi

LDA mencari batas pemisah kelas dengan asumsi distribusi tertentu. Metode ini kuat jika asumsi tersebut cukup sesuai dengan data. Ingat kembali fungsi diskriminan:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \log\pi_k, \quad (10.20)$$

di mana μ_k , Σ_k , dan π_k harus diestimasi, dan estimasinya diberikan oleh $\hat{\mu}_k$, $\hat{\Sigma}_k$, dan $\hat{\pi}_k$, masing-masing. Dengan menggunakan dekomposisi spektral dari $\hat{\Sigma}_k$ sebagai $\hat{\Sigma}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{U}_k^T$, di mana $\mathbf{D}_k = \text{diag}\{d_{k1}, \dots, d_{kp}\}$, fungsi diskriminan $\delta_k(x)$ dalam Persamaan (10.20) dapat diimplementasikan dalam bentuk data yang ditransformasikan. Secara khusus, suku $(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)$ dapat dinyatakan sebagai $(\mathbf{U}_k(x - \mu_k))^T \mathbf{D}_k^{-1} \mathbf{U}_k(x - \mu_k)$, dan determinan logaritmik $\log|\Sigma_k|$ disederhanakan menjadi $\sum_l \log d_{kl}$. Misalkan $\hat{\sigma}$ menyatakan estimasi matriks kovariansi umum, dan asumsikan dekomposisi spektralnya diberikan oleh $\hat{\sigma} = \mathbf{U} \mathbf{D} \mathbf{U}^T$. Transformasi data $\mathbf{X}$ ke $\mathbf{X}^* = \mathbf{D}^{-1/2} \mathbf{U}^T \mathbf{X}$ disebut sebagai “pembulatan”. Dalam ruang yang ditransformasikan ini, setiap observasi $\mathbf{X}^*$ ditugaskan ke centroid kelas terdekat, dengan memperhitungkan efek dari π_k .

LDA peringkat rendah memanfaatkan potensi representasi dimensi yang lebih rendah dari centroid K kelas. Ia menetapkan observasi ke kelas yang centroid-nya paling dekat dalam ruang dimensi yang lebih rendah ini. Proses ini melibatkan beberapa langkah. Pertama, hitung $K \times p$ matriks centroid $\mathbf{M}$ kelas dan matriks kovariansi umum dalam kelas $\mathbf{W}$. Kemudian, transformasikan centroid menggunakan $\mathbf{M}^* = \mathbf{M} \mathbf{W}^{-1/2}$, di mana transformasi tersebut memanfaatkan dekomposisi spektral dari $\mathbf{W}$. Selanjutnya, hitung $\mathbf{B}^*$, matriks kovariansi dari $\mathbf{M}^*$ (mewakili kovariansi antar kelas), dan lakukan dekomposisi spektralnya sebagai $\mathbf{B}^* =$



$\mathbf{V}^* \mathbf{D}_B \mathbf{V}^{*T}$. Kolom-kolom v_l^* dari $\mathbf{V}^*$ mendefinisikan koordinat subruang optimal secara berurutan.

Variabel diskriminan ke l -kemudian diberikan oleh $Z_l = v_l^T X$, di mana $v_l = \mathbf{W}^{-1/2} v_l^*$. Masalah Fisher dapat dinyatakan sebagai pencarian kombinasi linier yang $Z = a^T X$ memaksimalkan varians antar kelas relatif terhadap varians dalam kelas. Hal ini dapat dirumuskan sebagai:

$$\arg \max_a \frac{a^T \mathbf{B} a}{a^T \mathbf{W} a} \quad \text{or} \quad \arg \max_a \{a^T \mathbf{B} a : a^T \mathbf{W} a = 1\}, \quad (10.21)$$

yang solusinya adalah $a = v_1$, vektor eigen utama dari $\mathbf{W}^{-1} \mathbf{B}$. Demikian pula, vektor eigen berikutnya, seperti a_2 , dapat ditentukan dengan menyelesaikan

$$\arg \max_a \left\{ \frac{a^T \mathbf{B} a}{a^T \mathbf{W} a} : a^T \mathbf{W} a_1 = 0 \right\},$$

dan proses ini dapat dilanjutkan untuk mengekstrak komponen tambahan.

Catatan 10.2

Estimasi dan pengujian matriks kovariansi berdimensi tinggi tetap menjadi bidang pengembangan yang menantang dan aktif.

Catatan 10.3

Kita akan meninjau kembali konsep *hyperplane* pemisah dalam konteks *support vector machines* (SVM).

Perbandingan Antara Regresi Logistik dan LDA

LDA dengan distribusi bersyarat Gaussian memberikan

$$\begin{aligned} \log \frac{\Pr(G = k | X = x)}{\Pr(G = K | X = x)} \\ &= \log \frac{\pi_k}{\pi_K} - \frac{1}{2} (\mu_k + \mu_K)^T \sigma^{-1} (\mu_k - \mu_K) + x^T \sigma^{-1} (\mu_k - \mu_K) \\ &= \alpha_{k0} + \alpha_k^T x \end{aligned} \quad (10.22)$$

dan regresi logistik memberikan

$$\log \frac{\Pr(G = k | X = x)}{\Pr(G = K | X = x)} = \beta_{k0} + \beta_k^T x. \quad (10.23)$$

Mengingat

$$\Pr(X, G = k) = \Pr(X) \Pr(G = k | X) = \Pr(G = k) \Pr(X | G = k).$$

Jadi, untuk kedua metode tersebut,



$$\Pr(G = k | X = x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}, \quad k \neq K,$$

Dan

$$\Pr(G = K | X = x) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}.$$

Regresi logistik tidak menentukan parameter secara spesifik $\Pr(X)$, melainkan memperkirakan parameter dengan memaksimalkan likelihood bersyarat $\Pr(G = k | X)$, sedangkan LDA memperkirakan parameter dengan memaksimalkan likelihood gabungan.

$$\Pr(X, G = k) = \phi(X; \mu_k, \Sigma) \pi_k,$$

di mana $\Pr(X)$ tidak berperan.

Catatan 10.4

Klasifikasi berdasarkan kemungkinan bersyarat mungkin kehilangan efisiensi.

Catatan 10.5

Ingat kembali log-likelihood untuk regresi logistik

$$l(\beta) = \sum_{i=1}^N y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}),$$

untuk klasifikasi biner. Jika data dalam model regresi logistik dua kelas dapat dipisahkan secara sempurna oleh sebuah *hyperplane*, estimasi kemungkinan maksimum dari parameter tidak terdefinisi. Namun, dalam kasus ini, koefisien LDA untuk data yang sama akan terdefinisi dengan baik karena kemungkinan marginal tidak akan memungkinkan degenerasi ini.

$$g(X) = \mathbb{E}(Y | X),$$

di mana $Y \in \mathbb{R}$ dan $X \in \mathbb{R}^p$. Motivasi utamanya adalah untuk melampaui model linier, kita dapat mengusulkan:

$$f(X) = \sum_{m=1}^M \beta_m h_m(X),$$

di mana setiap $h_m: \mathbb{R}^p \rightarrow \mathbb{R}$ dan masing-masing adalah skalar.

Fungsi Polinomial Potongan

Fungsi polinomial potongan dimaksudkan untuk beradaptasi dengan fitur lokal data atau fungsi $\mathbb{E}(Y | X)$ yang mendasarinya. Matriks basisnya adalah,



$$\mathbf{H} = (h_i(X_j)),$$

Matriks $N \times M$ dan metode pencocokan kuadrat terkecil masih dapat dilakukan dalam pengaturan ini.

Definisi 10.1

Sebuah spline ordo M dengan knots $\xi_j, j = 1, \dots, K$ adalah polinomial sesepotong berordo M dan memiliki turunan kontinu hingga ordo $M - 2$.

Untuk menggunakan spline, seseorang harus menentukan urutan dan simpulnya, yang bukanlah hal yang mudah dilakukan.

Spline Penghalus

Spline membantu model mengikuti hubungan yang tidak lurus. Tantangannya adalah menjaga kurva tetap halus tanpa terlalu mengikuti noise. Penggunaan regularisasi memungkinkan pemilihan simpul secara otomatis. Misalnya,

$$\text{RSS}(f, \lambda) = \sum_{i=1}^N [y_i - f(x_i)]^2 + \lambda \int (f''(t))^2 dt, \quad (10.24)$$

di mana λ disebut parameter penghalus. Minimizer untuk Persamaan (10.24) adalah spline alami, yang dapat ditulis sebagai:

$$f(x) = \sum_{j=1}^N N_j(x) \theta_j.$$

Persamaan kriteria (10.24) disederhanakan menjadi:

$$\text{RSS}(\theta, \lambda) = (\mathbf{y} - \mathbf{N}\theta)^T (\mathbf{y} - \mathbf{N}\theta) + \lambda \theta^T \mathbf{\Omega}_N \theta,$$

di mana $\mathbf{N}(i, j) = N_j(x_i)$ dan $\Omega_N(j, k) = \int N_j''(t) N_k''(t) dt$. Selanjutnya, solusinya adalah

$$\hat{\theta} = (\mathbf{N}^T \mathbf{N} + \lambda \mathbf{\Omega}_N)^{-1} \mathbf{N}^T \mathbf{y},$$

dan spline yang dipasang adalah

$$\hat{f}_\lambda(x) = \sum_{j=1}^N N_j(x) \hat{\theta}_j.$$

Memilih Parameter Penghalusan

Misalkan $\hat{\mathbf{f}}$ adalah vektor N -nilai yang sesuai $\hat{f}(x_i)$ pada prediktor pelatihan x_i . Maka,



$$\begin{aligned}\hat{\mathbf{f}}_\lambda &= \mathbf{N}(\mathbf{N}^T \mathbf{N} + \lambda \mathbf{\Omega}_N)^{-1} \mathbf{N}^T \mathbf{y} \\ &= \mathbf{S}_\lambda \mathbf{y},\end{aligned}$$

di mana $\mathbf{S}_\lambda$ disebut matriks penghalus. Perhatikan bahwa $\mathbf{N}$ adalah matriks persegi berdimensi N . "Derajat kebebasan efektif" dari *spline* penghalus didefinisikan sebagai:

$$\text{df}_\lambda = \text{trace}(\mathbf{S}_\lambda). \quad (10.25)$$

Misalkan $\mathbf{K}$ adalah matriks yang tidak bergantung pada λ sedemikian sehingga $\mathbf{S}_\lambda = (\mathbf{I} + \lambda \mathbf{K})^{-1}$, yang dapat diturunkan dari identitas matriks Woodbury. Tuliskan dekomposisi spektral dari $\mathbf{S}_\lambda$ sebagai:

$$\mathbf{S}_\lambda = \sum_{k=1}^N \rho_k(\lambda) \mathbf{u}_k \mathbf{u}_k^T, \quad \text{dimana } \rho_k(\lambda) = \frac{1}{1 + \lambda d_k}, \quad (10.26)$$

dan d_k merupakan nilai eigen dari $\mathbf{K}$. Selanjutnya,

$$\mathbf{S}_\lambda \mathbf{y} = \sum_{k=1}^N \rho_k(\lambda) \mathbf{u}_k \langle \mathbf{u}_k, \mathbf{y} \rangle. \quad (10.27)$$

Kovariansi titiknya adalah

$$\text{Cov}(\hat{\mathbf{f}}_\lambda) = \mathbf{S}_\lambda \text{Cov}(\mathbf{y}) \mathbf{S}_\lambda^T,$$

yang entri diagonalnya berisi varians titik demi titik pada x_i . Biasanya adalah:

$$\text{Bias}(\hat{\mathbf{f}}_\lambda) = \hat{\mathbf{f}}_\lambda - \mathbf{S}_\lambda \mathbb{E}(\mathbf{y}).$$

Validasi silang adalah teknik untuk memilih parameter penghalusan. Secara spesifik, kurva validasi silang leave-one-out -fold didefinisikan sebagai:

$$\text{CV}(\hat{\mathbf{f}}_\lambda) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{f}_\lambda^{-i}(x_i))^2 = \frac{1}{N} \sum_{i=1}^N \left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_\lambda(i, i)} \right)^2 \quad (10.28)$$

Rumus Persamaan (10.28) sangat mudah.

Catatan 10.6



Karena kita tidak mengetahui kebenarannya, kita tidak mengetahui kesalahan prediksi kuadrat terintegrasi (EPE), bagaimana memperkirakan EPE merupakan topik riset yang sulit. Terdapat banyak makalah tentang validasi silang.

Ruang Hilbert

Sebenarnya, misalkan $\mathcal{L}^2(\mu)$ adalah ruang semua fungsi yang dapat diintegrasikan kuadrat terhadap ukuran sigma hingga μ . Maka $\mathcal{L}^2(\mu)$ adalah ruang Banach di bawah norma

$$\|h\|_2 = \left(\int |h(x)|^2 \mu(dx) \right)^{1/2},$$

dan memiliki hasil perkalian dalam:

$$\langle g, h \rangle = \int h(x)g(x) \mu(dx),$$

yang merupakan $\mathcal{L}^2(\mu)$ ruang Hilbert. Akibatnya, $\mathcal{L}^2(\mu)$ memiliki basis ortonormal lengkap $\{h_t: t \in T\}$ seperti:

$$f = \sum_{t \in T} \langle f, h_t \rangle h_t$$

- $\|\cdot\|_2$ untuk setiap $f \in \mathcal{L}^2(\mu)$. Ini adalah dasar untuk perluasan basis. Pertimbangkan masalah regularisasi:

$$\min_{f \in \mathcal{H}} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f),$$

di mana J adalah sebuah fungsi pada $\mathcal{H}$. Misalkan *kernel*nya dapat diperluas:

$$K(x, y) = \sum_{i=1}^{\infty} \gamma_i \phi_i(x) \phi_i(y), \quad (10.29)$$

Dengan $\gamma_i > 0$ dan $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$. Kemudian $\mathcal{H}$ dapat dipilih sebagai:

$$\mathcal{H}_K = \left\{ f: f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x) \right\},$$

dengan batasan

$$\|f\|_{\mathcal{H}_K}^2 = \sum_{i=1}^{\infty} \frac{c_i^2}{\gamma_i} < \infty.$$



Pengaturan $J(f) = \|f\|_{\mathcal{H}_K}^2$. Kemudian solusinya

$$\min_{f \in \mathcal{H}} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f),$$

dapat ditulis sebagai:

$$f(x) = \sum_{i=1}^N \alpha_i K(x, x_i),$$

dan oleh karena itu:

$$J(f) = \sum_{i=1}^N \sum_{j=1}^N K(x_i, x_j) \alpha_i \alpha_j,$$

Dengan kata lain, kita memiliki kriteria berdimensi terbatas:

$$\min_{\alpha} L(\mathbf{y}, \mathbf{K}\alpha) + \lambda \alpha^T \mathbf{K}\alpha \quad \text{for which } \mathbf{K}(i, j) = K(x_i, x_j).$$

Metode ini akan ditinjau kembali dalam SVM, yaitu,

$$\min_{\alpha_0, \alpha} \sum_{i=1}^N (1 - y_i f(x_i))_+ + \frac{\lambda}{2} \alpha^T \mathbf{K}\alpha.$$

Catatan 10.7

Sifat mereproduksi berasal dari $f(x) = \langle f, K_x \rangle$ dan $K_x(y) = K(x, y)$, di mana $K_x(\cdot) = K(x, \cdot)$. K harus kontinu, simetris, dan semi-definit positif, yang terakhir ini berarti

$$\sum_{i=1}^n \sum_{j=1}^n K(a_i, a_j) c_i c_j \geq 0,$$

untuk semua urutan titik $x_1, \dots, x_n$ (di dalam domain) dan semua pilihan $c_1, \dots, c_n$. Pikirkan tentang matriks semi-definit positif. Ekspansi $K(x, y) = \sum_{i=1}^{\infty} \gamma_i \phi_i(x) \phi_i(y)$ berkaitan dengan teorema Mercer.

Model Aditif Tergeneralisasi

Keluar dari kerangka kerja penggunaan fungsi basis yang telah ditentukan sebelumnya ke kerangka kerja semi/nonparametrik. Model



$$\mathbb{E}(Y | X_1, \dots, X_p) = \alpha + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p), \quad (10.30)$$

di mana $X_1, X_2, \dots, X_p$ adalah prediktor dan Y adalah hasilnya. Di sini f_i tidak ditentukan secara implisit. Bandingkan ini dengan pencocokan menggunakan fungsi basis:

$$RSS(\theta, \lambda) = (\mathbf{y} - \mathbf{N}\theta)^T(\mathbf{y} - \mathbf{N}\theta) + \lambda\theta^T\mathbf{\Omega}_N\theta$$

dengan pas

$$\hat{\mathbf{f}}_\lambda = \mathbf{N}(\mathbf{N}^T\mathbf{N} + \lambda\mathbf{\Omega}_N)^{-1}\mathbf{N}^T\mathbf{y} = \mathbf{S}_\lambda\mathbf{y}.$$

Contoh 10.1

Klasifikasi dua kelas. Misalkan $\mu(X) = \Pr(Y = 1 | X)$ dan

$$\log\left(\frac{\mu(X)}{1 - \mu(X)}\right) = \alpha + \beta_1 X_1 + \dots + \beta_p X_p,$$

versus

$$\log\left(\frac{\mu(X)}{1 - \mu(X)}\right) = \alpha + f_1(X_1) + \dots + f_p(X_p),$$

di mana setiap f_j diasumsikan mulus (*smooth*).

Secara umum, melalui fungsi penghubung g untuk $\mu = \mathbb{E}(Y | X)$, model aditif umum (GAM) dapat ditulis sebagai:

$$g[\mathbb{E}(Y | X)] = \alpha + f_1(X_1) + \dots + f_p(X_p), \quad (10.31)$$

misalnya, Gaussian dengan $g = \iota$, Poisson dengan $g = \log$, Binomial dengan $g = \text{logit}$. Alasan penggunaan fungsi penghubung adalah untuk mengakomodasi rentang GAM.

Contoh 10.2

$g(\mu) = X^T\beta + \alpha_k + f(Z)$, sebuah model semiparametrik, di mana X adalah vektor prediktor, α_k merupakan efek untuk level ke- k dari variabel kategoris V , dan efek dari prediktor Z dimodelkan secara nonparametrik.

Contoh 10.3

$g(\mu) = f(X) + g_k(Z)$, di mana k mengindeks level dari input kategoris V , dan dengan demikian menghasilkan suku interaksi $h(V, Z) = g_k(Z)$, sehingga efeknya tidak selalu bersifat aditif.



Catatan 10.8

Ada harga yang harus dibayar saat beralih dari kerangka kerja parametrik ke kerangka kerja semi/nonparametrik. Bayangkan Anda mencocokkan model semiparametrik untuk regresi linear sederhana dengan galat acak Normal yang bersifat i.i.d. (independen dan berdistribusi identik).

Memasang GAM

GAM menggabungkan beberapa fungsi halus dari fitur yang berbeda. Hasilnya lebih fleksibel daripada regresi linier biasa, tetapi masih cukup mudah dijelaskan. Ingat kembali modelnya

$$g[\mathbb{E}(Y | X)] = \alpha + f_1(X_1) + \dots + f_p(X_p)$$

Atau

$$Y = \alpha + f_1(X_1) + \dots + f_p(X_p) + \epsilon,$$

di mana ϵ memiliki nilai rata-rata nol. Algoritma yang digunakan untuk menyesuaikan GAM disebut *backfitting*. Ketika f_j merupakan *spline* kubik dalam komponen X_j , maka dimungkinkan untuk:

- Inisialisasi $\hat{\alpha} = \frac{1}{N} \sum_{i=1}^N y_i$ dan $\hat{f}_j \equiv 0, \forall i, j$
- Lakukan iterasi pada: $\hat{f}_j \leftarrow \mathbf{S}_j \left[\{y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(x_{ik})\}_{i=1}^N \right]$ dan

$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N} \sum_{i=1}^N \hat{f}_j(x_{ij}) \quad \left(\text{so that } \sum_{i=1}^N \hat{f}_j(x_{ij}) = 0 \text{ for all } j \right)$$

Catatan 10.9

Untuk regresi logistik (atau *penalized likelihood* lainnya), matriks Hessian terlibat dalam rutinitas Newton–Raphson, yang kemudian menginduksi kuadrat terkecil tertimbang secara iteratif (*iteratively reweighted least squares* atau IRLS).

Catatan 10.10

Paket R "GAM" mengimplementasikan model aditif tergeneralisasi (*generalized additive models*), dan begitu pula dengan "VGAM" (singkatan dari "*vector generalized linear and additive models*").

Ilustrasi: Memprediksi Spam Email

Berikut beberapa detail yang harus Anda perhatikan secara khusus:

- a. Tentukan derajat kebebasan untuk *spline*
- b. Transformasi log untuk observasi dengan distribusi berekor panjang



- c. Komponen linier adalah penyesuaian linier kuadrat terkecil tertimbang dari kurva yang dipasang pada prediktor, sedangkan bagian nonlinier adalah residual
- d. Penggunaan skor Z dalam uji signifikansi koefisien pada komponen linier
- e. Kerugian yang berbeda untuk kategori yang berbeda dalam klasifikasi

Regresi dan Klasifikasi Berbasis Pohon

Ide dasar di balik metode berbasis pohon adalah membagi domain menjadi subset dan memodelkan respons sebagai konstanta pada setiap subset, yaitu,

$$f(X) = \sum_{i \in \mathcal{G}} c_i \mathbf{1}_{R_i}(X), \quad \text{where } \mathcal{G} = \sum_{i \in \mathcal{G}} R_i.$$

Catatan 10.11

Setiap fungsi terukur (*measurable function*) dapat didekati dengan fungsi sederhana (*simple functions*). Jika fungsi tersebut terbatas (*bounded*), maka pendekatannya adalah seragam (*uniform*).

Pohon Regresi

Model

$$f(x) = \sum_{i=1}^M c_m I(x \in R_m). \quad (10.32)$$

Memasang pohon regresi dengan meminimalkan jumlah kuadrat mengarah pada estimasi

$$\hat{c}_m = \mathbf{ave}(y_i \mid x_i \in R_m). \quad (10.33)$$

Perhatikan bahwa $\hat{c}_m$ yang diberikan dalam Persamaan (10.33) diperoleh ketika partisi $\{R_m\}_{m=1}^M$ diketahui.

Catatan 10.12

Kesulitan juga muncul dari penentuan partisi optimal, yang memerlukan optimasi "bersama" (*joint optimization*). Skema yang disebutkan di atas mirip dengan metode tetangga terdekat (*nearest-neighbor methods*). Algoritma greedy beroperasi secara "koordinat" alih-alih "bersama-sama". "Pemangkasan kompleksitas biaya" mencari pohon yang optimal:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|, \quad (10.34)$$

di mana $N_m = |\{x_i \in R_m\}|$, $\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$, $Q_m(T) = \frac{1}{N_m}$, dan T adalah subpohon dari pohon T_0 . Sekali lagi, validasi silang digunakan untuk menentukan α .

Pohon Klasifikasi

Untuk mengklasifikasikan hasil yang mengambil nilai $1, 2, \dots, K$, kita dapat mendefinisikan sebagai:



$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k),$$

dan mengklasifikasikan pengamatan dalam mode m ke kelas.

$$\hat{k}(m) = \arg \max_k \hat{p}_{mk}.$$

Namun, kita masih memiliki bentuk kriteria yang sama.

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|,$$

seperti pada kasus regresi. Implementasi: (1) Tetapkan α , temukan T_α ; (2) lakukan validasi silang untuk memilih α untuk mendapatkan hasil akhir $\hat{\alpha}$ dan $T_{\hat{\alpha}}$. Tiga ukuran untuk mengukur ketidakmurnian node adalah:

- Kesalahan klasifikasi: $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m))$
- Indeks Gini: $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk})$
- Entropi silang atau devians: $-\sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$

Sebuah interpretasi dari indeks Gini: jika kita mengklasifikasikan pengamatan dalam sebuah simpul m ke kelas k dengan probabilitas $\hat{p}_{mk}$, maka ekspektasi kesalahan pelatihan dari aturan ini pada simpul tersebut adalah $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'}$. Demikian pula, jika kita mengodekan setiap pengamatan sebagai 1 untuk kelas k dan nol untuk lainnya, maka varians pada simpul tersebut adalah $\hat{p}_{mk}(1 - \hat{p}_{mk})$.

Tantangan dalam Metode Berbasis Pohon

Metode berbasis pohon, meskipun menawarkan wawasan yang berharga, menghadapi beberapa tantangan. Pertama, menangani prediktor kategoris dapat menuntut komputasi yang besar. Jumlah partisi potensial untuk variabel kategoris dengan q tingkatan tumbuh secara eksponensial, khususnya $2^{q-1} - 1$ kemungkinan cara untuk membaginya ke dalam dua kelompok. Kedua, pemilihan fungsi kerugian yang tepat secara signifikan memengaruhi performa model. Di dalam simpul terminal, aturan Bayes empiris menyarankan klasifikasi ke kelas $k(m) = \arg \min_k \sum_l L_{lk} \hat{p}_{ml}$, di mana L_{lk} mewakili kerugian yang timbul akibat mengklasifikasikan observasi dari kelas l ke dalam kelas k , dan $\hat{p}_{ml}$ menunjukkan estimasi probabilitas dari suatu observasi yang termasuk dalam kelas l di simpul m . Ketiga, pembagian multi-arah, meskipun berpotensi informatif, dapat menyebabkan ketidakcukupan data pada simpul-simpul berikutnya, sehingga menghambat estimasi model yang akurat. Selain itu, menangani nilai prediktor yang hilang memerlukan pertimbangan yang cermat.

Variabel pengganti, yang merupakan prediktor yang sangat berkorelasi dengan prediktor yang hilang, dapat digunakan untuk mengimputasi nilai yang hilang atau memandu



proses pembagian. Meskipun pohon berbobot dapat meningkatkan daya prediksi, hal ini sering kali mengorbankan interpretasi model. Selain itu, model berbasis pohon menunjukkan ketidakstabilan yang melekat. Kesalahan dalam pembagian awal dapat menyebar ke seluruh struktur pohon, memengaruhi pembagian selanjutnya dan pada akhirnya memengaruhi kinerja keseluruhan model. Ketidakstabilan yang melekat ini tidak dapat dihilangkan sepenuhnya. Terakhir, sifat konstan per bagian dari model berbasis pohon dapat membatasi kemampuannya untuk menangkap hubungan yang halus dan kontinu dalam data. Selain itu, mereka mungkin kesulitan untuk secara efektif memodelkan struktur aditif ketika hubungan mendasar yang sebenarnya melibatkan interaksi antara fitur yang membagi ruang fitur menjadi subset yang berpotongan.

Contoh Ilustratif: Prediksi Spam

Pilihan fungsi kerugian secara signifikan memengaruhi kinerja pengklasifikasi berbasis pohon, seperti yang dibuktikan oleh kurva karakteristik operasi penerima (ROC). Area di bawah kurva ROC memberikan metrik yang berharga untuk mengevaluasi kinerja pengklasifikasi dan setara dengan statistik Mann–Whitney U, yang mengukur perbedaan median dalam skor prediksi antara kedua kelompok.

Campuran Hierarkis Pakar dan Nilai yang Hilang

Campuran hierarkis pakar (HME) menentukan aliran observasi ke dalam suatu kelas melalui pemodelan probabilistik langsung. Ini adalah perbedaan utama dari CART, yang menggunakan "pemisahan keras". Namun, bayangkan kompleksitas masalah optimasi untuk HME. Misalkan y adalah vektor respons dan X adalah matriks input berukuran $N \times p$. Nyatakan dengan X_{obs} entri-entri terobservasi dari X dan misalkan $Z = (y, X)$ dan $Z_{\text{obs}} = (y, X_{\text{obs}})$. Misalkan R menjadi matriks indikator pada status hilangnya entri-entri dari X . Data dikatakan "hilang secara acak" jika

$$\Pr(R \mid Z, \theta) = \Pr(R \mid Z_{\text{obs}}, \theta),$$

di mana θ adalah parameter untuk distribusi R . "Hilang sepenuhnya secara acak" mengacu pada:

$$\Pr(R \mid Z, \theta) = \Pr(R \mid \theta).$$

Jika diasumsikan fitur-fitur hilang secara acak sepenuhnya, beberapa opsi tersedia:

- Buang observasi dengan nilai yang hilang
- Mengandalkan algoritma pembelajaran untuk menangani nilai yang hilang dalam fase pelatihannya; misalnya, pembagian pengganti
- Mengisi semua nilai yang hilang sebelum pelatihan

Catatan 10.13

CART dapat menjadi mesin imputasi yang baik. Korelasi antar fitur dapat membantu mengimputasi nilai yang hilang. Menggunakan nilai hasil imputasi sebagai observasi mengabaikan ketidakpastian akibat imputasi tersebut.



Penghalus Kernel Satu Dimensi

Metode kernel memberi bobot lebih besar pada data yang dekat dengan titik yang sedang diprediksi. Lebar kernel menentukan seberapa luas tetangga yang ikut memengaruhi hasil. Ingat bahwa pembelajaran terawasi dapat diartikan secara setara sebagai memperkirakan ekspektasi bersyarat $\mathbb{E}(Y | X)$, di mana Y adalah respons dan X prediktornya. Ditulis secara titik demi titik, kita memiliki

$$f(x) = \mathbb{E}(Y | X = x).$$

Intuisi dasarnya adalah memperkirakan f di suatu lokasi x dengan rata-rata tertimbang dari pengamatan di sekitarnya x . Dengan pengamatan (Y_i, X_i) , $i = 1, \dots, N$ untuk (Y, X) , cara paling sederhana adalah memperkirakan dengan $f(x)$:

$$\hat{f}(x) = \text{Ave}(y_i | x_i \in N_k(x)) = \frac{1}{|N_k(x)|} \sum_{x_i \in N_k(x)} y_i, \quad (10.35)$$

yaitu, memperkirakan $f(x)$ sebagai rata-rata dari y_i yang berada dalam lingkungan berukuran k dari x . Di sisi lain, pertimbangkan rata-rata tertimbang *kernel* Nadaraya–Watson, yang didefinisikan sebagai:

$$\hat{f}(x_0) = \frac{1}{\sum_{i=1}^N K_\lambda(x_0, x_i)} \sum_{i=1}^N K_\lambda(x_0, x_i) y_i \quad (10.36)$$

di mana kernel K dapat dipilih sebagai:

$$K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{\lambda}\right)$$

Dan

$$D(t) = \frac{3}{4}(1 - t^2)\mathbf{1}_{\{|t| \leq 1\}}(t).$$

Bentuk umum:

$$K_\lambda(x_0, x) = D\left(\frac{\|x - x_0\|}{h_\lambda(x_0)}\right)$$

Pertimbangan dalam Penghalusan Kernel

Keberhasilan penerapan penghalusan kernel bergantung pada beberapa faktor utama. Pertama, pemilihan fungsi kernel (K) yang bijaksana dan lebar pita terkaitnya (h_λ) sangatlah



krusial. Pilihan yang tidak tepat dapat secara signifikan memengaruhi akurasi dan interpretabilitas dari estimasi yang dihasilkan. Kedua, adanya nilai yang sama dalam variabel prediktor (x_i) memerlukan perhatian yang cermat. Salah satu pendekatan umum untuk mengatasi masalah ini melibatkan penggunaan "bobot observasi". Teknik ini memerlukan pembobotan ulang respons rata-rata pada setiap nilai X yang berpasangan, yang secara efektif memitigasi potensi bias yang dapat timbul dari observasi duplikat. Akhirnya, masalah efek batas harus dipertimbangkan dengan cermat. Karena sifat dari penghalusan kernel, estimator secara tidak sengaja dapat menetapkan massa positif di luar batas pendukung yang sebenarnya dari fungsi densitas yang mendasarinya. Hal ini dapat menyebabkan estimasi yang tidak akurat, terutama di dekat tepi distribusi data.

Regresi Lokal dan Metode Kemungkinan Lokal

Untuk penghalus kernel, teks tersebut menyatakan bahwa estimasi di dekat batas tersebut cenderung bias ke atas ketika "fungsi sebenarnya adalah linier di dekat batas" dan bahwa hal ini terjadi "karena asimetri kernel di wilayah tersebut." Regresi berbobot lokal akan melakukan koreksi urutan pertama terhadap bias tersebut. Regresi linier lokal:

$$\min_{\alpha(x_0), \beta(x_0)} \sum_{i=1}^N K_{\lambda}(x_0, x_i) [y_i - \alpha(x_0) - \beta(x_0)x_i]^2, \quad (10.37)$$

dan perkiraannya adalah

$$\hat{f}(x_0) = \alpha(x_0) + \beta(x_0)x_0. \quad (10.38)$$

Perhatikan bahwa Persamaan (10.37) adalah masalah kuadrat terkecil berbobot. Jadi,

$$\hat{f}(x_0) = b(x_0)^T (B^T W_{x_0} B)^{-1} B^T W_{x_0} y = \sum_{i=1}^N l_i(x_0) y_i, \quad (10.39)$$

untuk beberapa fungsi l_i . Perhatikan bahwa l_i disebut sebagai bobot kernel ekuivalen. Asumsikan ekspansi tersebut:

$$\begin{aligned} \mathbb{E}[\hat{f}(x_0)] &= \sum_{i=1}^N l_i(x_0) f(x_i) \\ &\approx f(x_0) \sum_{i=1}^N l_i(x_0) + f'(x_0) \sum_{i=1}^N l_i(x_0)(x_i - x_0) \\ &\quad + \frac{f''(x_0)}{2} \sum_{i=1}^N l_i(x_0)(x_i - x_0)^2. \end{aligned} \quad (10.40)$$



Kemudian, pada Persamaan (10.40), $f(x_0) \sum_{i=1}^N l_i(x_0) = f(x_0)$ karena $\sum_{i=1}^N l_i(x_0) = 1$, dan $f'(x_0) \sum_{i=1}^N l_i(x_0)(x_i - x_0) = 0$, karena $\sum_{i=1}^N l_i(x_0)(x_i - x_0) = 0$. Jadi,

$$\mathbb{E}[\hat{f}(x_0)] = f(x_0) + o(|x_i - x_0|).$$

Catatan 10.14

$\sum_{i=1}^N K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2 = \|W^{1/2}\mathbf{y} - W^{1/2}(1, \mathbf{x})(\alpha, \beta)\|$, di mana W adalah matriks diagonal dengan entri-entri diagonal $K_\lambda(x_0, x_i)$ dan $(1, \mathbf{x})$ adalah matriks $N \times 2$ yang mana $\mathbf{x} = (x_1, \dots, x_N)$. Regresi polinomial lokal:

$$\min_{\alpha(x_0), \beta_j(x_0)} \sum_{i=1}^N K_\lambda(x_0, x_i) \left[y_i - \alpha(x_0) - \sum_{j=1}^d \beta_j(x_0)x_i^j \right]^2.$$

Memilih Lebar Kernel

Ingat kembali ketiga kasus tersebut:

- Untuk kernel kompak dan halus: λ adalah radius pendukung.
- Untuk kernel halus dengan pendukung tak terbatas: λ biasanya merupakan kecepatan di mana kernel tersebut mendekati fungsi Dirac (dalam distribusi berdasarkan analisis fungsional).
- Untuk tetangga terdekat: λ adalah jumlah tetangga yang digunakan untuk membangun rata-rata lokal.

Peringatan: komentar dalam teks mengenai kompromi bias-varians bergantung pada kontinuitas dari f . Lebar kernel untuk regresi linear lokal biasanya ditentukan oleh validasi silang. Penghalus regresi lokal adalah penghalus linear, dan matriks penghalusnya adalah $\mathbf{S}_\lambda = (l_i(x_j))$, di mana λ adalah lebar pita dan bukan kekuatan penalti. Validasi silang dapat digunakan untuk menentukan λ .

Kernel Terstruktur dan Metode Kemungkinan Lokal

Misalkan $x \in \mathbb{R}^p$ dan ukuran sampel tidak N sebanding dengan p . Kita dapat menukar simetri radial untuk mendapatkan lebih banyak fleksibilitas (seperti pengurangan dimensi) sebagai berikut:

$$K_\lambda(x_0, x) = D \left(\frac{(x - x_0)^T \mathbf{A}(x - x_0)}{\lambda} \right),$$

di mana $\mathbf{A}$ adalah matriks semidefinit positif. Metode kemungkinan lokal:

$$l(\beta(x_0)) = \sum_{i=1}^N K_\lambda(x_0, x_i) g(y_i, x_i^T \beta(x_0)),$$

di mana fungsi kemungkinannya.



Estimasi Kepadatan Kernel

Pertimbangkan estimasi f_X kepadatan variabel acak X berdasarkan sampel acak $x_1, \dots, x_N$. Estimator titik dari $f_X(x_0)$ adalah:

$$\hat{f}_X(x_0) = \frac{|\{x_i \in \mathcal{N}_\lambda(x_0)\}|}{N\lambda},$$

di mana $\mathcal{N}_\lambda(x_0)$ lingkungan metrik x_0 di sekitar lebar λ . Perhatikan perbedaan di atas dari metode tetangga terdekat. Estimasi Parzen dengan kernel Gaussian adalah:

$$\hat{f}_X(x) = \frac{1}{N\lambda} \sum_{i=1}^N K_\lambda(x, x_i) = \frac{1}{N\lambda} \sum_{i=1}^N \phi_\lambda(x - x_i) = (\hat{F} * \phi_\lambda)(x),$$

di mana $\hat{F}(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{x_i \leq x\}}$ (ketika merupakan besaran skalar) dan

$$\phi_\lambda(x) = \frac{1}{\sqrt{2\pi\lambda}} \exp\left(-\frac{x^2}{2\lambda}\right).$$

Versi multivariatnya ϕ_λ adalah:

$$\phi_\lambda(x) = \frac{1}{(2\pi\lambda^2)^{p/2}} \exp\left(-\frac{1}{2} \frac{\|x\|^2}{\lambda^2}\right),$$

untuk $x \in \mathbb{R}^p$. Dalam kasus ini,

$$\hat{f}_X(x) = \frac{1}{N(2\pi\lambda^2)^{p/2}} \sum_{i=1}^N \exp\left(-\frac{1}{2} \frac{\|x_i - x\|^2}{\lambda^2}\right).$$

Teorema 10.1

Misalkan f dapat didiferensialkan secara kontinu sebanyak m kali dengan $\int |f^{(m)}(x)|^2 dx < \infty$. Asumsikan

$$\int (K_\lambda(y))^2 < \infty, \int |y|^m K_\lambda(y) dy < \infty$$

dan

$$\int K_\lambda(y) = 1, \int y K_\lambda(y) = 0, \dots, \int y^{m-1} K_\lambda(y) = 0.$$



Maka terdapat sebuah konstanta C_f sedemikian sehingga untuk $\lambda > 0$ yang kecil

$$\text{MISE}_f(\hat{f}) = \int \mathbb{E}_f \left(\hat{f}(x) - f(x) \right)^2 dx \leq C_f \left(\frac{1}{N\lambda} + \lambda^{2m} \right).$$

Akibatnya, untuk $\lambda \sim N^{-1/(2m+1)}$,

$$\text{MISE}_f(\hat{f}) = O(N^{-2m/(2m+1)}).$$

Penerapan pada Klasifikasi

Probabilitas posterior $\Pr(G = j \mid X = x_0)$ dapat didekati dengan:

$$\hat{q}_j(x_0) = \frac{\hat{\pi}_j \hat{f}_j(x_0)}{\sum_{k=1}^J \hat{\pi}_k \hat{f}_k(x_0)},$$

di mana $\hat{\pi}_j$ dan $\hat{f}_j(x_0)$ masing-masing mewakili probabilitas prior yang diestimasi dan densitas bersyarat kelas. Penting untuk dicatat bahwa untuk tugas klasifikasi, estimasi akurat densitas kelas penuh mungkin tidak diperlukan. Dalam kasus seperti itu, fokus dapat dialihkan ke estimasi probabilitas posterior di dekat batas keputusan, yang didefinisikan sebagai:

$$\left\{ x: \Pr(G = 1 \mid X = x) = \frac{1}{2} \right\},$$

untuk masalah klasifikasi biner.

Model Campuran

Model campuran Gaussian (GMM) didefinisikan sebagai:

$$f(x) = \sum_{m=1}^M \alpha_m \phi(x; \mu_m, \Sigma_m), \quad (10.41)$$

di mana $\phi(x; \mu_m, \Sigma_m)$ menyatakan fungsi kepadatan probabilitas dari distribusi Gaussian multivariat dengan rata-rata μ_m dan matriks kovariansi Σ_m . Koefisien pencampuran $(\alpha_1, \dots, \alpha_m)$ dibatasi untuk berada di dalam simpleks:

$$\mathcal{S} = \{x = (x_1, \dots, x_M) \in \mathbb{R}^M: x_i \geq 0, \|x\|_1 = 1\}.$$

Menariknya, ketika $\Sigma_m = \delta > 0$ untuk semua m dan jumlah komponen campuran M mendekati ukuran sampel N , GMM pada Persamaan (10.41) konvergen menjadi estimasi kepadatan kernel dengan $\alpha_m = \frac{1}{N}$ dan $\mu_m = x_m$, di mana x_m mewakili titik data ke- m . Teks tersebut memberikan aplikasi ilustratif dari GMM pada dataset penyakit jantung koroner.



10.3 KOMPLEKSITAS MODEL

Kesalahan prediksi pada sampel uji independen:

$$\text{Err}_{\mathcal{T}} = \mathbb{E} \left[L \left(Y, \hat{f}(X) \right) \mid \mathcal{T} \right]$$

di mana L adalah fungsi kerugian dan himpunan $\mathcal{T}$ data pelatihan bersifat tetap. Ini disebut "kesalahan generalisasi" atau "kesalahan pengujian". "Kesalahan prediksi yang diharapkan" atau "kesalahan pengujian yang diharapkan" adalah:

$$\text{Err} = \mathbb{E}[\text{Err}_{\mathcal{T}}].$$

“Kesalahan pelatihan”

$$\overline{\text{err}} = \frac{1}{N} \sum_{i=1}^N L \left(y_i, \hat{f}(x_i) \right).$$

Catatan 10.15

Galat pelatihan adalah versi empiris dari galat pengujian yang diharapkan. Mengapa galat pelatihan menurun seiring dengan kompleksitas model? Untuk Y kuantitatif, fungsi kerugian L dapat berupa kerugian kuadrat atau kerugian absolut. Untuk klasifikasi, kita mencoba mengestimasi $p_k(X) = \Pr(G = k|X)$ dan menetapkan X ke kelas di mana:

$$\hat{G}(X) = \arg \max_k \hat{p}_k(X).$$

Fungsi kerugian L dalam pengaturan ini dapat berupa:

$$L \left(G, \hat{G}(X) \right) = \mathbf{1} \left(G \neq \hat{G}(X) \right),$$

atau penyimpangan:

$$L \left(G, \hat{p}(X) \right) = -2 \sum_{k=1}^K \mathbf{1} (G = k) \log \hat{p}_k(X).$$

Sekali lagi, kesalahan pengujiannya adalah:

$$\text{Err}_{\mathcal{T}} = \mathbb{E} \left[L \left(G, \hat{G}(X) \right) \mid \mathcal{T} \right]$$

Ada dua tujuan: pemilihan model dan penilaian model. Namun, kita perlu memperkirakan kesalahan-kesalahan ini untuk mencapai tujuan tersebut. Ketika kita memiliki data yang cukup, kita dapat membagi data menjadi tiga bagian: set pelatihan, set validasi, dan set pengujian.



Dekomposisi Bias-varians

Pertimbangkan model linier $Y = f(X) + \epsilon$ dengan $\mathbb{E}[\epsilon] = 0$ dan $V[\epsilon] = \sigma_\epsilon^2$. Dengan kerugian kuadrat,

$$\text{Err}(x_0) = \sigma_\epsilon^2 + \left(\mathbb{E}[\hat{f}(x_0)] - f(x_0) \right)^2 + \mathbb{E} \left[\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)] \right]^2.$$

Untuk regresi k -NN,

$$\text{Err}(x_0) = \sigma_\epsilon^2 + \left(f(x_0) - \frac{1}{k} \sum_{x_j \in \mathcal{N}} f(x_j) \right)^2 + \frac{\sigma_\epsilon^2}{k},$$

di mana kita telah mengasumsikan observasi i.i.d. dan bahwa x_i bersifat deterministik. Untuk kecocokan model linear $\hat{f}(x) = \langle x, \beta \rangle$ di mana $\beta \in \mathbb{R}^p$ diestimasi dengan kuadrat terkecil,

$$\text{Err}(x_0) = \sigma_\epsilon^2 + \left(\mathbb{E}[\hat{f}(x_0)] - f(x_0) \right)^2 + \| \mathbf{h}(x_0) \|^2 \sigma_\epsilon^2,$$

di mana $\mathbf{h}(x) = X(X^T X)^{-1}x$ karena $\hat{f}(x_0) = x_0^T (X^T X)^{-1} X^T y$ dan kesalahan acak tidak berkorelasi. Apakah benar bahwa dekomposisi di atas hanya berlaku untuk kerugian kuadrat? Apakah Anda setuju bahwa “Secara umum, kesalahan pengujian yang diharapkan dapat diuraikan menjadi bias model dan bias estimasi?” Kesalahan “**dalam sampel**” adalah:

$$\frac{1}{N} \sum_{i=1}^N \text{Err}(x_i) = \sigma_\epsilon^2 + \frac{1}{N} \sum_{i=1}^N \left(\mathbb{E}[\hat{f}(x_i)] - f(x_i) \right)^2 + \frac{p}{N} \sigma_\epsilon^2,$$

sejak

$$\begin{aligned} V[x_0^T (X^T X)^{-1} X^T y] &= \text{tr}\{x_0^T (X^T X)^{-1} X^T V[y] X (X^T X)^{-1} x_0\} \\ &= \text{tr}(x_0^T (X^T X)^{-1} x_0) \end{aligned}$$

dan

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \mathbb{V}[x_i^T (X^T X)^{-1} X^T y] &= \frac{1}{N} \sum_{i=1}^N \|\mathbf{h}(x_i)\|^2 = \frac{1}{N} \sum_{i=1}^N \text{tr}(x_i^T (X^T X)^{-1} x_i) \\ &= \frac{1}{N} \text{tr}((X^T X)^{-1} X X^T). \\ &= \frac{p}{N} \end{aligned}$$



Memperkirakan Kesalahan

Diberikan kumpulan data pelatihan:

$$\mathcal{T} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

dan sebuah titik $(X^0, Y^0) \in \mathcal{T}$ dan $(X^0, Y^0) \sim F$, kesalahan generalisasi suatu model $\hat{f}$ adalah:

$$\text{Err}_{\mathcal{T}} = \mathbb{E}_{(X^0, Y^0)} [L(Y^0, \hat{f}(X^0)) | \mathcal{T}]$$

dan kesalahan yang diharapkan adalah:

$$\text{Err} = \mathbb{E}_{\mathcal{T}} \mathbb{E}_{(X^0, Y^0)} [L(Y^0, \hat{f}(X^0)) | \mathcal{T}]$$

Perlu dicatat bahwa sebagian besar metode secara efektif memperkirakan Err daripada $\text{Err}_{\mathcal{T}}$.
Ingat kembali kesalahan pelatihan

$$\overline{\text{err}} = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i))$$

dan kesalahan dalam sampel

$$\text{Err}_{\text{in}} = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{Y_i^0} [L(Y_i^0, \hat{f}(x_i)) | \mathcal{T}]$$

di mana Y^0 menunjukkan bahwa N nilai respons baru diamati pada x_i , $i = 1, \dots, N$.

Catatan 10.17

$\overline{\text{err}}$ biasanya lebih kecil dari $\text{Err}_{\mathcal{T}}$ dan merupakan risiko empiris dalam buku V. Vapnik. Optimisme rata-rata didefinisikan sebagai:

$$\omega = \mathbb{E}_{\mathbf{y}} [\text{Err}_{\text{in}} - \overline{\text{err}}],$$

di mana prediktor ditetapkan dalam definisi sebelumnya dan ekspektasinya adalah atas nilai respons set pelatihan. Biasanya kita memiliki

$$\omega = \frac{2}{N} \sum_{i=1}^N \text{Cov}(\hat{y}_i, y_i)$$

Dan



$$\mathbb{E}_{\mathbf{y}}[\text{Err}_{\text{in}}] = \mathbb{E}_{\mathbf{y}}[\overline{\text{err}}] + \frac{2}{N} \sum_{i=1}^N \text{Cov}(\hat{y}_i, y_i). \quad (10.42)$$

Identitas penting ini mendasari beberapa kriteria pemilihan variabel. Ketika diperoleh melalui penyesuaian linier dengan input atau fungsi basis,

$$\sum_{i=1}^N \text{Cov}(\hat{y}_i, y_i) = d\sigma_{\epsilon}^2, \quad (10.43)$$

(yang merupakan dasar untuk jumlah parameter yang efektif) dan

$$\mathbb{E}_{\mathbf{y}}[\text{Err}_{\text{in}}] = \mathbb{E}_{\mathbf{y}}[\overline{\text{err}}] + \frac{2}{N} d\sigma_{\epsilon}^2. \quad (10.44)$$

Kriteria informasi Akaike (AIC),

$$\text{AIC} = -\frac{2}{N} \log L(\hat{\theta}; X, Y) + 2 \frac{d}{N}.$$

Hal ini didasarkan pada

$$-2\mathbb{E}[\log L(\hat{\theta}; X, Y)] \approx -\frac{2}{N} \mathbb{E} \left[\sum_{i=1}^N \log L(\hat{\theta}; X, Y_i) \right] + 2 \frac{d}{N}, \quad (10.45)$$

sebagai $N \rightarrow \infty$. Ada kesalahan ketik pada identitas (7.27) dalam teks. Kriteria informasi Bayesian (BIC),

$$\text{BIC} = -2\log L(\hat{\theta}; X, Y) + d\log N.$$

Hal ini berkaitan dengan faktor Bayes:

$$\frac{\Pr(\mathcal{M}_m | \mathbf{Z})}{\Pr(\mathcal{M}_l | \mathbf{Z})} = \frac{\Pr(\mathcal{M}_m) \Pr(\mathbf{Z} | \mathcal{M}_m)}{\Pr(\mathcal{M}_l) \Pr(\mathbf{Z} | \mathcal{M}_l)},$$

dan berdasarkan pada

$$\log \Pr(\mathbf{Z} | \mathcal{M}_m) = \log \Pr(\mathbf{Z} | \hat{\theta}_m, \mathcal{M}_m) - \frac{d_m}{2} \log N + O(1),$$

di mana $\mathbf{Z}$ berisi data pelatihan dan $\Pr(\mathcal{M}_m)$ merupakan konstanta. Perhatikan bahwa:



$$\Pr(\mathcal{M}_m | \mathbf{Z}) \propto \Pr(\mathcal{M}_m) \int \Pr(\mathbf{Z} | \theta_m, \mathcal{M}_m) \Pr(\theta_m | \mathcal{M}_m) d\theta_m.$$

Dengan kata lain, memilih model dengan BIC minimum sama artinya dengan memilih model dengan probabilitas posterior (perkiraan) terbesar. BIC cenderung memberikan hukuman yang lebih berat pada model yang kompleks dibandingkan AIC. BIC secara asimtotik konsisten, tetapi AIC mungkin tidak.

Catatan 10.18

Terdapat banyak hasil mengenai AIC dan BIC, dalam paradigma klasik dan paradigma modern.

Validasi Silang

Validasi silang membantu menilai model dengan beberapa pembagian data. Teknik ini memberi gambaran performa yang lebih stabil daripada satu kali pembagian saja. Estimasi validasi silang dari kesalahan prediksi adalah

$$CV(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha))$$

dan $CV(\hat{f})$ hanya memperkirakan kesalahan rata-rata

$$\text{Err} = \mathbb{E}_{\mathcal{T}} \mathbb{E}_{(X^0, Y^0)} [L(Y^0, \hat{f}(X^0)) | \mathcal{T}] = \mathbb{E} [L(Y, \hat{f}(X))]$$

Perlu dicatat bahwa dalam kondisi tertentu,

$$\mathbb{E} [L(Y, \hat{f}(X))] \approx \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i)) = \overline{\text{err}}.$$

Catatan 10.19

Untuk validasi silang K -lipat dengan ukuran sampel N . Ketika $K = N$, estimator validasi silang mendekati tidak bias untuk galat prediksi yang sebenarnya (diharapkan), tetapi dapat memiliki varians yang tinggi karena N "sampel pelatihan" sangat mirip satu sama lain. Validasi silang umum (GCV). Recall

$$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y},$$

Untuk banyak model, kita telah membahasnya di bawah kerugian kesalahan kuadrat. GCV untuk validasi silang *leave-one-out* adalah

$$\text{GCV}(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^N \left(\frac{y_i - \hat{f}(x_i)}{1 - N^{-1} \text{trace}(\mathbf{S})} \right)^2.$$



Prinsip untuk menerapkan validasi silang: model yang akan divalidasi harus independen dari model yang disarankan oleh data validasi.

Bootstrap

Lakukan sampel ulang dengan pengembalian, lakukan analisis berdasarkan masing-masing himpunan data hasil sampel ulang, dan kemudian rangkum hasilnya. Misalkan $\mathbf{Z} = (z_1, \dots, z_N)$, dengan $z_i = (x_i, y_i)$ sebagai himpunan data pelatihan. Misalkan $S(\mathbf{Z})$ adalah suatu kuantitas yang dihitung dari $\mathbf{Z}$. Lakukan sampel ulang dari $\{z_1, \dots, z_N\}$ dengan pengembalian untuk mendapatkan sampel bootstrap $\mathbf{Z}^{*b}$, dan hitung $S(\mathbf{Z}^{*b})$. Ambil $\{S(\mathbf{Z}^{*b})\}_{b=1}^B$ seolah-olah mereka diperoleh dari B himpunan data pelatihan baru. Estimasi kegagalan prediksi

$$\widehat{\text{Err}}_{\text{boot}} = \frac{1}{B} \frac{1}{N} \sum_{b=1}^B \sum_{i=1}^N L(y_i, \hat{f}^{*b}(x_i)).$$

Catatan 10.20

Validasi silang menggunakan data yang tidak tumpang tindih, tetapi bootstrap memiliki peluang positif untuk selalu menggunakan data yang tumpang tindih. Bootstrap dengan metode leave-one-out:

$$\widehat{\text{Err}}^{(1)} = \frac{1}{N} \sum_{i=1}^N \sum_{b \in C^{-i}} \frac{1}{|C^{-i}|} L(y_i, \hat{f}^{*b}(x_i)).$$

Algoritma EM

Contoh klasik estimasi campuran normal. $Y_1 \sim N(\mu_1, \sigma_1^2)$ dan $Y_2 \sim N(\mu_2, \sigma_2^2)$ dan $Y \sim N(\mu_2, \sigma_2^2)$, di mana $\Delta \in \{0,1\}$. Dengan kata lain,

$$Y \sim (1 - \Delta)Y_1 + \Delta Y_2.$$

Parameter yang akan diestimasi adalah $\theta = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$. Diberikan sampel acak $\{y_i\}_{i=1}^N$ untuk Y , fungsi *log-likelihood* adalah:

$$l(\theta, \mathbf{Z}) = \sum_{i=1}^N \ln [(1 - \pi)\phi(y_i; \mu_1, \sigma_1^2) + \pi\phi(y_i; \mu_2, \sigma_2^2)].$$

Perhatikan bahwa hal ini $l(\theta, \mathbf{Z})$ memiliki bentuk yang rumit karena kita tidak mengetahui keanggotaan untuk setiap observasi. Pertimbangkan estimasi θ melalui metode kemungkinan maksimum, yaitu,

$$\hat{\theta} \in \arg \max_{\theta \in \Theta} l(\theta, \mathbf{Z}).$$



Komplikasi dari optimasi langsung $l(\theta, \mathbf{Z})$ disebabkan oleh logaritma dari suatu penjumlahan, yaitu $[(1 - \pi)\phi(y_i; \mu_1, \sigma_1^2) + \pi\phi(y_i; \mu_2, \sigma_2^2)]$. suku-sukunya. Misalkan keanggotaan setiap pengamatan diketahui, maka

$$l(\theta, \mathbf{Z}, \Delta) = \sum_{i=1}^N [(1 - \Delta_i) \ln \phi(y_i; \mu_1, \sigma_1^2) + \Delta_i \ln \phi(y_i; \mu_2, \sigma_2^2)] \quad (10.46)$$

$$+ \sum_{i=1}^N [(1 - \Delta_i) \ln(1 - \pi) + \Delta_i \ln \pi],$$

di mana $\Delta = (\Delta_1, \dots, \Delta_N)$. Ekspresi Persamaan (10.46) tepatnya adalah:

$$l(\theta, \mathbf{Z}, \Delta) = \sum_{i=1}^N \ln [\phi(y_i; \mu_1, \sigma_1^2)^{1-\Delta_i} (1 - \pi)^{1-\Delta_i} \phi(y_i; \mu_2, \sigma_2^2)^{\Delta_i} \pi^{\Delta_i}].$$

Perhatikan bahwa diperoleh dengan menambahkan data dengan status keanggotaan $l(\theta, \mathbf{Z}, \Delta)$ yang tidak diketahui. Bandingkan dengan $l(\theta, \mathbf{Z})$, kita sudah melihat jejak algoritma MM dari Hunter dan Lange (2004) yang disebutkan dalam Contoh 8.7 pada teks. Ketika Δ diketahui, MLE untuk π adalah proporsi di $\{\Delta_i\}_{i=1}^N$ antara $\Delta_i = 1$. Namun, Δ tidak diketahui. Jadi,

$$\gamma_i(\theta) = \mathbb{E}[\Delta_i | \theta, \mathbf{Z}] = \Pr(\Delta_i = 1 | \theta, \mathbf{Z}), \quad (10.47)$$

dapat digunakan untuk mengestimasi kemungkinan y_i berasal dari anggota kedua. Ini adalah langkah ekspektasi. Gantikan Δ_i dalam Persamaan (10.46) dengan Persamaan (10.47) dan maksimalkan $l(\theta, \mathbf{Z}, \gamma_i(\theta))$ memberikan estimasi dari θ . Ini adalah langkah maksimisasi.

Dua Interpretasi Lain dari Algoritma EM

Misalkan $\mathbf{Z}$ adalah data yang diamati, $\mathbf{Z}^m$ data yang tidak diamati, dan $\mathbf{T} = (\mathbf{Z}, \mathbf{Z}^m)$ data yang diperluas. Perhatikan identitas:

$$\Pr(\mathbf{Z} | \theta') = \frac{\Pr(\mathbf{T} | \theta')}{\Pr(\mathbf{Z}^m | \theta', \mathbf{Z})}.$$

Maka log likelihood memenuhi syarat:

$$l(\theta'; \mathbf{Z}) = l_0(\theta'; \mathbf{T}) - l_1(\theta'; \mathbf{Z}^m | \mathbf{Z}) \quad (10.48)$$

dan



$$\begin{aligned} l(\theta'; \mathbf{Z}) &= \mathbb{E}_{\mathbf{T}|\mathbf{Z}}[l_0(\theta'; \mathbf{T})] - \mathbb{E}_{\mathbf{T}|\mathbf{Z}}[l_1(\theta'; \mathbf{Z}^m | \mathbf{Z})] \\ &= \mathbb{E}_{Q(\tilde{\theta}', \theta)}[l_0(\theta'; \mathbf{T}) | (\theta, \mathbf{Z})] - \mathbb{E}_{R(\tilde{\theta}', \theta)}[l_1(\theta'; \mathbf{Z}^m | \mathbf{Z}) | (\theta, \mathbf{Z})] \end{aligned}$$

Berdasarkan ketidaksetaraan Jensen:

$$\begin{aligned} R(\theta', \theta) &= \mathbb{E}_{\mathbf{T}|\mathbf{Z}}[l_1(\theta'; \mathbf{Z}^m | \mathbf{Z})] \\ &= \mathbb{E}_{\mathbf{T}|\mathbf{Z}}[\ln f_1(\theta'; \mathbf{Z}^m | \mathbf{Z})] \end{aligned}$$

Nilainya dimaksimalkan ketika $\theta^* = \theta$. Hal ini disebabkan oleh:

$$\int \ln \left(\frac{f(x)}{g(x)} \right) g(x) dx \leq \ln \int \frac{f(x)}{g(x)} g(x) dx = \ln 1 = 0,$$

yaitu,

$$\int \ln h(x) \mu(dx) \leq \ln \int h(x) \mu(dx) = \ln 1, \quad (10.49)$$

di mana $g \geq 0$ dan $\int g = 1$ menyiratkan bahwa $\mu(dx) = g(x)dx$ adalah ukuran hingga, $f \geq 0$ dan $\int f = 1$ menyiratkan bahwa $h = \frac{f}{g} \geq 0$ hampir pasti hingga, dan $\varphi = \ln$ adalah cekung. Kesamaan terakhir dalam Persamaan (10.49) didasarkan pada teorema Radon–Nikodym. Ini juga menyiratkan

$$\int [\ln f(x)]g(x) dx \leq \int [\ln g(x)]g(x) dx.$$

singkatnya,

$$l(\theta^*; \mathbf{Z}) = Q(\theta^*, \theta) - R(\theta^*, \theta)$$

dan

$$l(\theta'; \mathbf{Z}) - l(\theta; \mathbf{Z}) = \underbrace{[Q(\theta', \theta) - Q(\theta, \theta)]}_{\geq 0} - \underbrace{[R(\theta', \theta) - R(\theta, \theta)]}_{\leq 0} \geq 0$$

jika memaksimalkan $Q(\theta^*, \theta)$. Jadi, iterasi EM tidak pernah mengurangi *log-likelihood*, dan cukup untuk mencapai:

$$\max_{\tilde{\theta} \in \Theta} Q(\tilde{\theta}, \hat{\theta}^{(j)}) = Q(\hat{\theta}^{(j+1)}, \hat{\theta}^{(j)}) \geq Q(\hat{\theta}^{(j)}, \hat{\theta}^{(j)}),$$

alih-alih memaksimalkan sepenuhnya $l(\theta; \mathbf{Z})$. Perspektif kedua. Pertimbangkan hal berikut:



$$\begin{aligned} F(\theta', \tilde{P}) &= \mathbb{E}_{\tilde{P}}[l_0(\theta'; \mathbf{T})] - \mathbb{E}_{\tilde{P}}[\ln \tilde{P}(\mathbf{Z}^m)] \\ &= \mathbb{E}_{\tilde{P}}[\ln \Pr(\mathbf{Z}|\theta') + \ln \Pr(\mathbf{Z}^m|\theta', \mathbf{Z})] - \mathbb{E}_{\tilde{P}}[\ln \tilde{P}(\mathbf{Z}^m)] \\ &= \mathbb{E}_{\tilde{P}}[\ln \Pr(\mathbf{Z}|\theta')] + \mathbb{E}_{\tilde{P}}[\ln \Pr(\mathbf{Z}^m|\theta', \mathbf{Z})] - \mathbb{E}_{\tilde{P}}[\ln \tilde{P}(\mathbf{Z}^m)], \end{aligned}$$

di mana $P(\mathbf{Z}^m)$ terdapat distribusi apa pun pada data laten. Perhatikan bahwa

$$F(\theta', \Pr(\mathbf{Z}^m | \theta', \mathbf{Z})) = l(\theta'; \mathbf{Z}) = \mathbb{E}_{\mathbf{T}|\mathbf{Z}}[l_0(\theta'; \mathbf{T})] - \mathbb{E}_{\mathbf{T}|\mathbf{Z}}[l_1(\theta'; \mathbf{Z}^m | \mathbf{Z})] \quad (10.50)$$

Lebih jauh,

$$\operatorname{argmax}_{\tilde{P}} F(\theta, \tilde{P}) = \Pr(\mathbf{Z}^m | \theta, \mathbf{Z}). \quad (10.51)$$

Ini adalah distribusi yang diberikan oleh langkah E. Pada langkah M, kita peroleh $\mathbb{E}_{\mathbf{T}|\mathbf{Z}}[l_1(\theta'; \mathbf{Z}^m | \mathbf{Z})]$. Perhatikan bahwa tidak bergantung pada kapan $\Pr(\mathbf{T} | \mathbf{Z}) = \Pr(\mathbf{Z}^m | \theta', \mathbf{Z})$. Jadi, cukup untuk memaksimalkan

$$\mathbb{E}_{\tilde{P}}[l_0(\theta'; \mathbf{T}) | (\theta, \mathbf{Z})] = \mathbb{E}_{\mathbf{T}|\mathbf{Z}}[l_0(\theta'; \mathbf{Z})]$$

berkaitan dengan θ . Perspektif ini terhubung dengan Bayes variasional.

10.4 ALGORITMA BAYESIAN

Pendekatan Bayesian memadukan informasi awal dan data baru. Hasilnya tidak hanya satu angka, tetapi gambaran ketidakpastian terhadap parameter atau prediksi. Kita akan membahas Bayes variasional dan komputasi Bayesian aproksimasi (ABC).

Bayes Variasional

Referensi utama adalah dan. Bayes variasional adalah metode untuk inferensi aproksimasi. Metode ini dapat digunakan baik dalam paradigma frequentist maupun Bayesian. Pertimbangkan inferensi Bayesian berdasarkan posterior

$$p(\theta | y) = \frac{p(y, \theta)}{p(y)},$$

di mana $p(y)$ letak kerapatan marginal untuk dan $p(y, \theta)$ kerapatan gabungan untuk Y dan θ . Kita perlu mengetahui $p(y)$ hal ini untuk melakukan inferensi Bayes. Namun,

$$p(y) = \int p(y, \theta) d\theta,$$

dan menghitung integralnya bisa jadi tidak memungkinkan. Jadi, salah satu caranya adalah dengan melakukan aproksimasi.



Identitas Kunci

Beberapa identitas:

$$\begin{aligned}\log p(y) &= (\log p(y)) \int q(\theta) d\theta \\ &= \int q(\theta) \log \left(\frac{p(y, \theta)}{q(\theta)} \right) d\theta + \int q(\theta) \log \left(\frac{p(\theta|y)}{q(\theta)} \right) d\theta, \\ &\geq \int q(\theta) \log \left(\frac{p(y, \theta)}{q(\theta)} \right) d\theta\end{aligned}$$

sejak

$$\int q(\theta) \log \left(\frac{p(\theta|y)}{q(\theta)} \right) d\theta \leq 0. \quad (10.52)$$

yaitu,

$$p(y) \geq \tilde{p}(y; q) = \exp \left(\int q(\theta) \log \left(\frac{p(y, \theta)}{q(\theta)} \right) d\theta \right). \quad (10.53)$$

Sekarang, tugasnya berubah menjadi:

$$\operatorname{argmax}_{q \in \mathcal{Q}} \tilde{p}(y; q) = \operatorname{argmax}_{q \in \mathcal{Q}} \exp \left(\int q(\theta) \log \left(\frac{p(y, \theta)}{q(\theta)} \right) d\theta \right) \quad (10.54)$$

Atau

$$\operatorname{argmax}_{q \in \mathcal{Q}} L(q; p(\theta|y)) = \operatorname{argmax}_{q \in \mathcal{Q}} \int q(\theta) \log \left(\frac{p(\theta|y)}{q(\theta)} \right) d\theta. \quad (10.55)$$

Catatan 10.21

Bandinkan hal di atas dengan $\operatorname{argmax}_{\tilde{p}} F(\theta', \tilde{p}) = \Pr(\mathbf{Z}^m | \theta', \mathbf{Z})$ yang diberikan dalam Persamaan (10.51). Kita memilih q yang mudah dikelola.

Inferensi Variasional

Salah satu pilihan $\mathcal{Q}$ adalah himpunan sedemikian q sehingga q dapat difaktorkan, yaitu, $q(\theta) = \prod_{i=1}^M q_i(\theta_i)$ yang mana $\{\theta_i\}_{i=1}^M$ dapat digabungkan menjadi θ . Dalam hal ini,



$$\begin{aligned}
 \log \tilde{p}(y; q) &= \int q(\theta) \log \left(\frac{p(y, \theta)}{q(\theta)} \right) d\theta \\
 &= \int \prod_{i=1}^M q_i(\theta_i) \left\{ \log p(y, \theta) - \sum_{i=1}^M \log q_i(\theta_i) \right\} d\theta_1 \cdots d\theta_M \\
 &= \int q_1(\theta_1) \left\{ \prod_{i \neq 1} q_i(\theta_i) \log p(y, \theta) \right\} d\theta_1 - \int q_1(\theta_1) \log q_1(\theta_1) d\theta_1 \\
 &\quad + A_{\setminus \{-1\}}
 \end{aligned} \tag{10.56}$$

Membiarkan

$$\tilde{p}(y; \theta_1) \propto \exp \left(\int q_1(\theta_1) \left\{ \prod_{i \neq 1} q_i(\theta_i) \log p(y, \theta) \right\} d\theta_1 \right),$$

sedemikian rupa sehingga $\int \tilde{p}(y; \theta_1) d\theta_1 dy = 1$. Kemudian,

$$\log \tilde{p}(y; q) = \int q_1(\theta_1) \log \left(\frac{\tilde{p}(y; \theta_1)}{q_1(\theta_1)} \right) d\theta_1 + A_{(-1)}. \tag{10.57}$$

Jika kita melakukan optimasi terhadap $q_1(\theta_1)$, maka pengoptimalnya adalah:

$$q_1^*(\theta_1) = \tilde{p}(\theta_1 | y).$$

Pada dasarnya, inferensi variasional di bawah divergensi Kullback–Leibler dapat diringkas dalam satu kalimat:

$$\hat{f} = \operatorname{argmax}_{f \in \mathcal{F}} \int f(x) \log \left(\frac{q(x, z)}{f(x)} \right) dx.$$

Catatan 10.22

Pilihan divergensi dan himpunan layak menentukan inferensi. Sebagai contoh, untuk g yang diberikan, seseorang dapat melakukan

$$\hat{f} = * \operatorname{argmax}_{f \in \mathcal{F}} D(f, g).$$

Terdapat hubungan yang mendalam antara divergensi pada ruang ukuran peluang dan geometri dari ruang ukuran peluang; lihat, contohnya. Divergensi KL juga berkaitan dengan uji rasio kemiripan, lihat, contohnya.

Perbaikan dan Varian

Terdapat beberapa masalah dengan prototipe Bayes variasional: (1) informasi tentang ketergantungan antar parameter sebagian hilang ketika kerapatan yang dapat difaktorkan



digunakan untuk aproksimasi; (2) optimasi pada setiap langkah siklik mungkin tidak memiliki solusi eksplisit jika keluarga kerapatan tidak memiliki struktur aljabar yang cocok; (3) optimasi pada setiap langkah siklik mungkin masih sangat intensif, yang memotivasi gradien stokastik untuk menghemat sumber daya komputasi atau gradien alami untuk menghormati metrik, misalnya, metrik Riemannian, pada himpunan distribusi probabilitas.

Komputasi Bayesian Aproksimasi

ABC terkait dengan metode terima-tolak; referensi. Karena itu, ABC telah digunakan oleh Markov Chain Monte Carlo (MCMC).

Versi Diskrit

Pertimbangkan Y bahwa mengambil paling banyak nilai yang dapat dihitung. Diberikan yang diamati y , untuk setiap $i = 1, \dots, N$, ulangi hal berikut:

- Hasilkan dari distribusi prior
- Hasilkan z dari kemungkinan $f(\cdot | \theta')$, ulangi ini

sampai $z = y$, kemudian tetapkan $\theta_i = \theta'$. Pengamatan kuncinya adalah bahwa θ_i berasal dari distribusi bersama:

$$f(\theta_i) = \sum_{z \in \mathcal{Z}} \pi(\theta_i) f(z | \theta_i) \mathbf{1}_{\{z=y\}} = \pi(\theta_i) f(y | \theta_i) = g(y, \theta_i) \propto \pi(\theta_i | y). \quad (10.58)$$

Versi Kontinu

Misalkan $\mathcal{Z}$ adalah rentang dari Y . Diberikan nilai yang diamati y , untuk setiap $i = 1, \dots, N$, ulangi hal berikut:

- hasilkan dari distribusi prior $\pi(\cdot)$
- hasilkan z dari kemungkinan $f(\cdot | \theta')$, ulangi ini sampai $\rho(\eta(z), \eta(y)) \leq \varepsilon$, lalu tetapkan Pengamatan kuncinya adalah θ_i bahwa kira-kira berasal dari distribusi gabungan:

$$\pi_\varepsilon(\theta, z | y) = \frac{\pi(\theta) f(z | \theta) \mathbf{1}_{A_{\varepsilon, y}}(z)}{\int_{A_{\varepsilon, y}} \pi(\theta) f(z | \theta) dz d\theta'}. \quad (10.59)$$

di mana

$$A_{\varepsilon, y} = \{z \in \mathcal{Z} : \rho(\eta(z), \eta(y)) \leq \varepsilon\}. \quad (10.60)$$

jadi,

$$\begin{aligned} \pi_\varepsilon(\theta | y) &= \int \pi_\varepsilon(\theta, z | y) dz \propto \int \pi(\theta) f(z | \theta) \mathbf{1}_{A_{\varepsilon, y}}(z) dz \\ &\approx \pi(\theta) f(y | \theta) \end{aligned} \quad (10.61)$$

$$\propto \pi(\theta | y). \quad (10.62)$$



Isu

Masalah utamanya adalah pemilihan statistik ringkasan η , ukuran kemiripan ρ , dan toleransi ε . Pilihan yang diinginkan untuk η adalah statistik cukup untuk y . Namun, sangat sulit untuk menemukan statistik cukup. Terdapat banyak ρ yang tersedia, masing-masing dapat memberikan hasil yang berbeda. Toleransi ε juga memengaruhi biaya komputasi dan akurasi inferensi. Salah satu idenya adalah mengambil ε sebagai parameter dan mensimulasikannya dari sebuah prior.

10.5 JARINGAN SARAF

Jaringan saraf dapat dianggap sebagai prosedur dua langkah: mentransformasikan masukan dan kemudian mentransformasikan masukan yang telah ditransformasikan. Deskripsi matematis diberikan sebagai berikut:

- Masukan: $X_1, \dots, X_p$
- Mentransformasikan masukan: $\eta_m = \eta_m(X)$, misalnya, $\eta_m(X) = \alpha_{0m} + \alpha_m^T X$, untuk $m = 1, \dots, M$
- Variabel laten atau unit tersembunyi: $Z_m = \sigma(\eta_m)$ untuk $m = 1, \dots, M$, untuk suatu fungsi skalar σ . Sering kali σ mendekati suatu fungsi indikator, misalnya, $\sigma(t) = \frac{1}{1+e^{-t}}$
- Mentransformasikan variabel laten: $T_k = h_k(Z)$, di mana $Z = (Z_1, \dots, Z_M)$ dan h_k dapat bersifat linear dalam Z , misalnya, $h_k(Z) = \beta_{0k} + \beta_k^T Z$
- Keluaran: $f_k(X) = g_k(T)$ untuk $k = 1, \dots, K$, di mana $T = (T_1, \dots, T_K)$

Memasang Jaringan Saraf

Ketika η_m dan h_k bersifat linier, himpunan lengkap bobot dilambangkan dengan θ , dan terdiri dari

$$\begin{cases} \{\alpha_{0m}, \alpha_m\}_{m=1}^M & M(p+1) \text{ weights} \\ \{\beta_{0k}, \beta_k\}_{k=1}^K & K(M+1) \text{ weights} \end{cases}$$

Untuk regresi, tujuannya adalah jumlah kuadrat kesalahan:

$$R(\theta) = \sum_{k=1}^K \sum_{i=1}^N (y_{ik} - f_k(x_i))^2,$$

sedangkan untuk klasifikasi, yang digunakan adalah squared error atau cross entropy:

$$R(\theta) = - \sum_{k=1}^K \sum_{i=1}^N y_{ik} \log f_k(x_i),$$



dan pengklasifikasinya adalah $G(x) = \underset{k}{\operatorname{argmax}} f_k(x)$. Ketika dan linear h_k digunakan untuk klasifikasi, model jaringan saraf adalah model regresi logistik linear pada unit tersembunyi Z_m . Pendekatan umum untuk meminimalkan adalah melalui penurunan gradien, yang disebut "propagasi balik" dalam konteks ini. Lihat persamaan dalam teks. Secara khusus,

$$\begin{cases} \beta_{km}^{(r+1)} = \beta_{km}^{(r)} - \gamma_r \sum_{i=1}^N \frac{\partial R_i}{\partial \beta_{km}^{(r)}} \\ \alpha_{ml}^{(r+1)} = \alpha_{ml}^{(r)} - \gamma_r \sum_{i=1}^N \frac{\partial R_i}{\partial \alpha_{ml}^{(r)}} \end{cases}, \quad (10.63)$$

di mana γ_r "tingkat pembelajaran" dan "persamaan back-propagation" berada

$$\begin{cases} \frac{\partial R_i}{\partial \beta_{km}} = \delta_{ki} z_{mi} \\ \frac{\partial R_i}{\partial \alpha_{ml}} = s_{mi} x_{il} \end{cases} \quad (10.64)$$

di mana

$$\begin{cases} \delta_{ki} = 2(y_{ik} - f_k(x_i)) g'_k(\beta_k^T z_i) \\ s_{mi} = \sigma'(\alpha_m^T x_i) \sum_{k=1}^K \beta_{km} \delta_{ki} \end{cases} \quad (10.65)$$

Algoritma ini menggunakan estimasi saat ini $\hat{\theta}^{(r)}$ untuk memperoleh $\hat{f}_k^{(r)}(x_i)$, kemudian $\delta_{ki}^{(r)}$ dan kemudian $s_{mi}^{(r)}$, dan kemudian $\beta_{km}^{(r+1)}$ dan $\alpha_{ml}^{(r+1)}$. "Dalam algoritma propagasi balik, setiap unit tersembunyi meneruskan dan menerima informasi hanya ke dan dari unit-unit yang berbagi sambungan. Oleh karena itu, ini dapat diimplementasikan secara efisien pada komputer arsitektur paralel." Laju pembelajaran γ_r untuk pembelajaran daring adalah sebuah bentuk aproksimasi stokastik yang dibahas oleh, yang konvergensinya dijamin jika:

$$\gamma_r \rightarrow 0, \quad \sum_r \gamma_r = \infty \quad \text{and} \quad \sum_r \gamma_r^2 < \infty.$$

Beberapa Masalah dengan Jaringan Saraf Tiruan

Ringkasan satu kalimat: Model umumnya memiliki parameter berlebih; masalah optimasi bersifat non-konveks dan tidak stabil:

- Nilai Awal dan Tangan Tak Terlihat dari Fungsi Sigmoid:** Model biasanya berevolusi dari yang kira-kira linier menjadi sangat non-linier. Namun, algoritma dapat macet ketika bobot nol yang tepat digunakan untuk memulai algoritma.



- b. **Overfitting:** Terlalu banyak bobot menyebabkan *overfitting*. Regularisasi dapat diterapkan.
- c. **Skala Input:** Skala input memengaruhi bobot di lapisan bawah. Disarankan untuk menggunakan input yang terpusat dan terstandarisasi.
- d. Jumlah unit dan lapisan tersembunyi.
- e. **Beberapa Minimum:** Ketika fungsi tujuan memiliki beberapa optimum lokal, solusi akhir sering kali bergantung pada pilihan nilai awal. Rekomendasinya adalah untuk mendapatkan beberapa jaringan menggunakan nilai awal yang berbeda dan kemudian merata-ratakan prediksi di atas kumpulan jaringan sebagai prediksi akhir.

10.6 MESIN VEKTOR PENDUKUNG

SVM mencari batas keputusan dengan margin yang lebar. Margin membantu model lebih tahan terhadap variasi data. Mesin vektor pendukung dapat dianggap sebagai generalisasi dari pengklasifikasi dengan batas keputusan linier.

Bidang Pemisah

Data pelatihan: N pasangan (x_1, y_1) sampai (x_N, y_N) dengan $x_i \in \mathbb{R}^p$ dan $y_i \in \{1, -1\}$. Definisikan sebuah hiperbidang dengan

$$\{x: f(x) = x^T \beta + \beta_0 = 0\},$$

di mana β adalah vektor satuan yaitu, $\|\beta\| = 1$. Hal ini menghasilkan aturan klasifikasi.

$$G(x) = \text{sgn}(x^T \beta + \beta_0).$$

Bidang hiper dengan margin terbesar, jika ada, didefinisikan sebagai:

$$\max_{\beta_0, \beta, \|\beta\|=1} M \quad \text{subject to} \quad \min_{1 \leq i \leq N} y_i (x_i^T \beta + \beta_0) \geq M,$$

atau bentuk yang setara:

$$\min_{\beta, \beta_0} 2^{-1} \|\beta\|^2 \quad \text{subject to} \quad \min_{1 \leq i \leq N} y_i (x_i^T \beta + \beta_0) \geq 1.$$

Rumusan umumnya adalah:

$$\min |\beta| \quad \text{subject to} \quad \begin{cases} y_i (x_i^T \beta + \beta_0) \geq 1 - \xi_i, \forall i \\ \xi_i \geq 0, \sum_i \xi_i \leq \lambda \end{cases},$$

di mana λ adalah konstanta. Rumusan ini menyatakan bahwa titik-titik yang berada jauh di dalam batas kelasnya tidak memainkan peran penting dalam membentuk batas tersebut.



Vektor Pendukung

SVM mencari batas keputusan dengan margin yang lebar. Margin membantu model lebih tahan terhadap variasi data. Menyelesaikan masalah optimasi menggunakan pengali Lagrange:

$$L = \frac{1}{2} \|\beta\|^2 + C \sum_i \xi_i - \sum_i \alpha_i [y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)] - \sum_i \mu_i \xi_i,$$

memberikan kondisi orde pertama:

$$\hat{\beta} = \sum_{i=1}^N \alpha_i y_i x_i; \quad \sum_i \alpha_i y_i = 0; \quad \alpha_i = C - \mu_i, \quad (10.66)$$

serta kendala positif $\alpha_i, \mu_i, \xi_i \geq 0$. Kondisi Karush–Kuhn–Tucker (KKT) mensyaratkan:

$$\begin{cases} \alpha_i \{y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)\} = 0 \\ \mu_i \xi_i = 0 \\ y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \end{cases} \quad (10.67)$$

Dalam Persamaan (10.66), $\hat{\alpha}_i \neq 0$ untuk pengamatan i di mana kendala-kendalanya:

$$y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i,$$

terpenuhi. Pengamatan tersebut disebut "vektor pendukung." Fungsi keputusannya adalah:

$$\hat{G}(x) = \text{sgn}(x^T \hat{\beta} + \hat{\beta}_0).$$

Pembelajaran terawasi biasanya bertujuan untuk menangkap hubungan $f(y|x)$ berdasarkan sampel $(x_1, y_1), \dots, (x_N, y_N)$ pelatihan melalui beberapa fungsi kerugian $L(y, \hat{y})$. Seringkali hal ini berbentuk estimasi distribusi bersyarat $\Pr(Y | X)$, yang mana $\Pr(Y | X)$ tidak menjadi perhatian langsung. Pembelajaran tak terawasi bertujuan untuk menangkap secara langsung kepadatan gabungan $\Pr(X)$ berdasarkan N pengamatan dari $(x_1, \dots, x_N)$ dari sebuah vektor- p acak X . Contoh metode mencakup metode-metode yang mengidentifikasi manifold berdimensi rendah di dalam ruang- X atau wilayah yang mengandung modus dari $\Pr(X)$.

10.7 ANALISIS KLASTER

Analisis kluster mengelompokkan data tanpa label. Hasilnya perlu ditafsirkan dengan konteks domain karena algoritma hanya melihat kemiripan berdasarkan fitur yang diberikan. Inti dari analisis kluster adalah gagasan tentang tingkat kesamaan antara objek individu yang dikelompokkan.



- a. **Matriks kedekatan:** diberikan $\{x_i\}_{i=1}^N$, sebuah matriks kedekatan $D = (d_{ij})_{N \times N}$ dapat didefinisikan dan digunakan oleh algoritma pengelompokan.
- b. **Ketidakmiripan:** diberikan pengukuran $x_i = (x_{i1}, \dots, x_{ip})$ untuk $i = 1, \dots, N$, sebuah ketidakmiripan d_j antara atribut j ke- dapat didefinisikan dan kemudian

$$D(x_i, x_{i'}) = \sum_{j=1}^p d_j(x_{ij}, x_{i'j}),$$

dapat digunakan sebagai ketidakmiripan antara objek dan. Perhatian khusus harus diberikan pada variabel ordinal dan variabel kategorikal.

- c. **Ketidakmiripan objek:**

$$D(x_i, x_{i'}) = \sum_{j=1}^p w_j d_j(x_{ij}, x_{i'j}) \quad \text{with} \quad \min_{1 \leq j \leq p} w_j \geq 0 \quad \text{and} \quad \sum_{j=1}^p w_j = 1.$$

- d. **Rata-rata ketidakmiripan antar objek:**

$$\bar{D} = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N D(x_i, x_{i'}) = \sum_{j=1}^p w_j \bar{d}_j,$$

di mana

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N d_j(x_{ij}, x_{i'j}).$$

Jika $d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$ lalu $\bar{d}_j = 2 \times \text{var}_j$, di mana estimasi sampel dari atribut tersebut?

Algoritma Pengelompokan: Kombinatorial

Menetapkan setiap observasi $\{x_i\}_{i=1}^N$ ke salah satu kluster $K (< N)$ melalui pengkodean $C: \{1, \dots, N\} \rightarrow \{1, \dots, K\}$. Kerugian alami adalah:

$$W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'}).$$

Penguraian



$$T = \frac{1}{2} \sum_{i=1}^N \sum_{i'=1}^N d(x_i, x_{i'}) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \left(\left(\sum_{C(i')=k} + \sum_{C(i') \neq k} \right) d(x_i, x_{i'}) \right)$$

$$= \underbrace{\frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'})}_{W(C)} + \underbrace{\frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i') \neq k} d(x_i, x_{i'})}_{B(C)},$$

memberikan kriteria yang setara untuk optimasi $B(C)$. Jumlah penugasan yang berbeda adalah:

$$S(N, K) = \frac{1}{K!} \sum_{k=1}^K (-1)^{K-k} \binom{K}{k} k^N.$$

Misalnya, $S(19, 4) \approx 10^{10}$, yaitu beban komputasi yang sangat besar.

Algoritma Pengelompokan: k -rata-rata

k -rata-rata:

$$W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} \|x_i - x_{i'}\|^2$$

$$= \frac{1}{2} \sum_{k=1}^K N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2,$$

di mana rata-rata sampel observasi dalam kluster k ? Masalah optimasinya adalah:

$$\min_{C, \{m_k\}_{k=1}^K} \sum_{k=1}^K N_k \sum_{C(i)=k} \|x_i - m_k\|^2.$$

Catatan 10.23

Algoritma rata-rata k tepat digunakan ketika ukuran ketidakmiripan yang diambil adalah kuadrat jarak Euclidean, dan algoritma ini membutuhkan jumlah kluster yang telah ditentukan sebelumnya. Jumlah kluster yang berbeda dapat menghasilkan kluster yang berbeda secara drastis.

Algoritma Pengelompokan: Pengelompokan Hierarkis

Pengelompokan hierarkis tidak memerlukan jumlah kluster yang telah ditentukan sebelumnya, tetapi memerlukan ukuran ketidakmiripan antar kelompok. Algoritma ini memiliki dua paradigma dasar: "aglomeratif" (dari bawah ke atas) dan "divisif" (dari atas ke bawah). Sebagian besar metode aglomeratif dan beberapa metode divisif memiliki sifat monoton jika dilihat dari bawah ke atas. Hal ini memotivasi "*dendrogram*," di mana pohon



biner sedemikian rupa sehingga tinggi setiap simpul sebanding dengan nilai ketidakmiripan antar kelompok antara kedua anaknya. Namun, *dendrogram* harus dilihat terutama sebagai deskripsi struktur pengelompokan data sebagaimana yang ditentukan oleh algoritma tertentu yang digunakan. Pengelompokan aglomeratif dapat diimplementasikan melalui:

- Hubungan tunggal (SL): $d_{SL}(GH) = \min_{i \in G, i' \in H} d_{ii'}$
- Hubungan lengkap (CL): $d_{CL}(GH) = \max_{i \in G, i' \in H} d_{ii'}$
- Rata-rata kelompok (GA): $d_{GA}(GH) = \frac{1}{|G||H|} \sum_{i \in G, i' \in H} d_{ii'}$

Catatan 10.24

Ketidakmiripan rata-rata kelompok adalah estimasi dari

$$\iint d(x, x') p_G(x) p_H(x') dx dx'.$$

Pengelompokan hierarkis memiliki dua fitur: (1) pemotongan *dendrogram* pada berbagai ketinggian, menghasilkan jumlah cluster yang berbeda, dan himpunan cluster tersebut saling bersarang; (2) memberikan beberapa informasi pengurutan parsial tentang sampel.

Komponen Utama, Kurva, dan Permukaan

Komponen utama dari sekumpulan data di $\mathbb{R}^p$ menyediakan urutan pendekatan linear terbaik terhadap data tersebut, di antara semua peringkat $q \leq p$. Salah satu perspektifnya adalah sebagai berikut. Diberikan pengamatan $\{x_i\}_{i=1}^N$, pertimbangkan model linear peringkat q :

$$f(\lambda) = \mu + \mathbf{V}_q \lambda, \quad (10.68)$$

di mana $\mathbf{V}_q$ memiliki kolom ortonormal dan $\lambda \in \mathbb{R}^q$. Pencocokan kuadrat terkecil pada Persamaan (10.68) menghasilkan:

$$\min_{\mu, \{\lambda_i\}, \mathbf{V}_q} \|x_i - \mu - \mathbf{V}_q \lambda_i\|^2,$$

yang titik kritisnya memenuhi

$$\begin{cases} \hat{\mu} &= \bar{x} \\ \hat{\lambda}_i &= \mathbf{V}_q^T (x_i - \bar{x}) \end{cases}$$

Oleh karena itu, kita perlu menyelesaikan

$$\min_{\mathbf{V}_q} \| (x_i - \bar{x}) - \mathbf{V}_q \mathbf{V}_q^T (x_i - \bar{x}) \|^2.$$

Ketika $\bar{x} = 0$, solusi dapat diperoleh dari SVD



$$\mathbf{X}_{N \times p} = \mathbf{U}_{N \times p} \mathbf{D}_{p \times p} \mathbf{V}^T = \sum_{i=1}^{\text{rank}(\mathbf{X})} d_i \mathbf{u}_i \mathbf{v}_i^T,$$

di mana $\mathbf{V}_q$ terdiri dari q kolom utama dari $\mathbf{V}$. Perhatikan bahwa $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ dan $\mathbf{V}^T \mathbf{V} = \mathbf{I}$. Kolom-kolom dari $\mathbf{U} \mathbf{D}$ disebut "komponen utama dari $\mathbf{X}$." $\hat{\lambda}_i = \mathbf{V}_q^T \mathbf{x}_i$ yang optimal, yaitu, $\hat{\lambda}_i$ adalah baris ke- i dari $\mathbf{U}_q \mathbf{D}_q$.

Catatan 10.25

Ingat kembali latihan tentang mencari subruang optimal dengan kendala peringkat di bawah norma Frobenius. Hal ini berkaitan erat dengan *analisis komponen utama* (PCA). Perspektif kedua: temukan vektor-vektor yang saling ortogonal $\mathbf{v}_j$ sedemikian sehingga $\|\mathbf{X} \mathbf{v}_j\|$ dimaksimalkan secara berturut-turut, yaitu, $\mathbf{v}_1^* = \arg\max_{\mathbf{v}} \|\mathbf{X} \mathbf{v}\|$, $\mathbf{v}_2^* = \arg\max_{\mathbf{v}} \{\|\mathbf{X} \mathbf{v}\| : \langle \mathbf{v}, \mathbf{v}_1^* \rangle = 0\}$ dan seterusnya. Pesan utamanya adalah "subset yang relatif kecil dari komponen utama dapat menjadi fitur berdimensi lebih rendah yang sangat baik untuk merepresentasikan data berdimensi tinggi." Contoh pada digit tulisan tangan. Setiap citra adalah sebuah titik $\mathbf{x}_i \in \mathbb{R}^{256}$. Model dua komponen yang dihasilkan adalah:

$$f(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2.$$

Contoh pada angka tulisan tangan. Setiap gambar adalah sebuah titik. Model dua komponen yang diinduksi adalah:

$$f(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2.$$

Transformasi Procrustes dan Perataan Bentuk

Transformasi ini dapat digunakan untuk "membandingkan" gambar yang direpresentasikan oleh matriks. Perhatikan masalah berikut:

$$\min_{\mu \in \mathbb{R}^p, R \in O^q} \|\mathbf{X}_2 - (\mathbf{X}_1 R + \mathbf{1} \mu^T)\|_F,$$

di mana $\|\mathbf{A}\|_F^2 = \text{trace}(\mathbf{A}^T \mathbf{A})$. Misalkan $\bar{x}_1$ dan $\bar{x}_2$ adalah sentroid kolom dari $\mathbf{X}_1$ dan $\mathbf{X}_2$, dan $\tilde{\mathbf{X}}_1$ dan $\tilde{\mathbf{X}}_2$ secara berurutan, adalah versi terpusat kolom dari $\mathbf{X}_1$ dan $\mathbf{X}_2$. Maka SVD $\tilde{\mathbf{X}}_1^T \tilde{\mathbf{X}}_2 = \mathbf{U} \mathbf{D} \mathbf{V}^T$ memberikan solusi

$$\begin{cases} \hat{\mathbf{R}} &= \mathbf{U} \mathbf{V}^T \\ \hat{\mu} &= \bar{x}_2 - \hat{\mathbf{R}} \bar{x}_1 \end{cases}$$

Jarak minimal disebut sebagai "jarak Procrustes." "Rata-rata Procrustes" dari sekumpulan bentuk L didefinisikan sebagai:



$$\min_{\mathbf{M}, \mathbf{R}_l \in \mathcal{O}} \sum_{l=1}^L \|\mathbf{X}_l \mathbf{R}_l - \mathbf{M}\|_F^2.$$

Secara lebih umum, kita dapat mendefinisikan rata-rata bentuk yang "invarian affin" melalui

$$\min_{\mathbf{M}, \mathbf{R}_l \in \mathcal{O}} \sum_{l=1}^L \|\mathbf{X}_l \mathbf{R}_l - \mathbf{M}\|_F^2.$$

Solusi dari persamaan di atas adalah: $\mathbf{M}$ merupakan matriks berukuran N kali p yang dibentuk oleh p vektor eigen terbesar dari

$$H = \frac{1}{L} \sum_{l=1}^L X_l (X_l^T X_l)^{-1} X_l^T.$$

Model Faktor dan Analisis Komponen Independen

SVD:

$$\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T,$$

memiliki representasi variabel laten: $\mathbf{S} = \sqrt{N} \mathbf{U}$ dan $\mathbf{A}^T = \mathbf{D} \mathbf{V}^T / \sqrt{N}$. Jadi, $\mathbf{X}$ setiap kolom dari merupakan kombinasi linear dari kolom-kolom $\mathbf{S}$. Perhatikan bahwa representasi

$$\mathbf{X} = \mathbf{S} \mathbf{A}^T,$$

unik modulo matriks ortogonal $\mathbf{R}$, yaitu, jika diuraikan seperti $\mathbf{X}$ di atas, maka solusinya unik $(\mathbf{S}, \mathbf{A})$ hingga transformasi ortogonal. Dengan perspektif ini, PCA dapat dianggap sebagai estimasi model variabel laten.

$$X = AS.$$

Sekarang perhatikan model faktor.

$$X_i = \sum_{j=1}^q a_{ij} s_j + \varepsilon_i, \quad i = 1, \dots, p$$

Atau

$$X = \mathbf{A} \mathbf{S} + \varepsilon \quad \text{with} \quad \mathbf{S} = (s_1, \dots, s_q)^T \quad \text{and} \quad q < p. \quad (10.69)$$



Kemudian kita memiliki apa yang disebut “model faktor” atau “model variabel laten”: $\mathbf{A}$ disebut “loading,” dan variabel laten atau faktor. Model Persamaan (10.69) menyiratkan struktur kovariansi sebagai

$$\Sigma = \mathbf{A}\mathbf{A}^T + \mathbf{D}_\varepsilon.$$

Analisis Komponen Independen

Asumsikan s_j saling independen dalam model $X = \mathbf{A}S + \varepsilon$, kita memperoleh model analisis komponen independen (ICA). Seringkali diasumsikan bahwa $\text{Cov}(X) = \mathbf{I}$ dan $\text{Cov}(S) = \mathbf{I}$. Hal ini memaksa $\mathbf{A}$ menjadi ortogonal ketika $q = p$. Kriteria yang digunakan untuk ICA dapat didasarkan pada entropi atau divergensi Kullback–Leibler.

Kurva Utama dan Skala Multidimensi

Diberikan variabel acak $X \in \mathbb{R}^p$ dan kurva halus parametrik $f: T \rightarrow \mathbb{R}^p$, misalkan $\lambda_f(x)$ adalah titik terdekat pada kurva ke $X = x$. Maka $f(\lambda)$ disebut kurva utama untuk distribusi X jika.

$$f(\lambda) = E(X \mid \lambda_f(X) = \lambda). \quad (10.70)$$

Ini berarti bahwa pengurangan dimensi yang diinduksi oleh kurva utama adalah melalui parameter berdimensi rendahnya. Perhatikan λ_f bahwa mengidentifikasi nilai parameter λ untuk setiap x . Konsep kurva utama dapat digeneralisasikan ke permukaan utama.

Penskalaan multidimensi (MDS) bertujuan untuk menemukan representasi data berdimensi rendah yang mempertahankan jarak antar pasangan sebaik mungkin. Dengan diberikan pengamatan $x_1, \dots, x_N \in \mathbb{R}^p$, tujuannya adalah untuk menemukan nilai $z_1, z_2, \dots, z_N \in \mathbb{R}^k$ untuk meminimalkan

$$S_M(z_1, \dots, z_N) = \sum_{i \neq i'} (d_{ii'} - \|z_i - z_{i'}\|)^2,$$

di mana $d_{ii'}$ adalah ketidaksamaan antara observasi x_i dan $x_{i'}$. Reduksi dimensi nonlinier berguna untuk masalah di mana rasio sinyal-terhadap-derau sangat tinggi. MDS klasik tidak mempertahankan urutan titik pada manifold, sedangkan MDS lokal mempertahankannya.

10.8 MODEL GRAFIS

Model grafis menampilkan hubungan antarvariabel sebagai simpul dan sisi. Bentuk ini membantu pembaca melihat struktur ketergantungan secara lebih visual. Kita akan membahas secara singkat pengujian berganda, model grafis, dan struktur eigen dari matriks kovarians sampel.

Tingkat Penemuan Palsu

Dalam statistik modern, sering kali banyak hipotesis diuji secara bersamaan, misalnya, dalam studi ekspresi gen diferensial, studi pencitraan otak, genetika, dan lain-lain. Misalkan



ada hipotesis nol H_{i0} , $i = 1, \dots, m$ sedemikian rupa sehingga m_0 di antaranya adalah benar dan pengujian salah. Jika prosedur pengujian $\mathcal{T}$ menolak hipotesis nol yang benar, maka telah terjadi penemuan palsu. Misalkan V adalah jumlah penemuan palsu dan R jumlah penolakan. Dalam statistik klasik, ukuran kesalahan untuk menyeimbangkan kesalahan tipe I keseluruhan dan daya dalam pengaturan ini adalah tingkat kesalahan keluarga. Ini didefinisikan sebagai:

$$\alpha' = \Pr(V \geq 1).$$

Konsep yang relatif baru untuk tujuan yang sama adalah "tingkat penemuan palsu" yang diusulkan oleh Benjamini dan Hochberg pada tahun 1995. Konsep ini didefinisikan sebagai:

$$\alpha(\mathcal{F}) = \mathbb{E} \left(\frac{V}{R \vee 1} \right) \quad (10.71)$$

Konsep false discovery rate (FDR) memiliki pengaruh besar pada inferensi statistik. Konsep ini dapat diterapkan pada sejumlah besar pengujian. FDR akan berkurang menjadi FWER ketika semua hipotesis nol benar.

Catatan 10.27

Definisi nilai- p memiliki analogi Bayesian sebagai nilai- q .

Biasanya, setiap hipotesis nol H_{i0} dikaitkan dengan sebuah nilai- p , p_i , dan pengujian hipotesis biasanya didasarkan pada nilai- p yang terdistribusi secara seragam di bawah hipotesis nol. Metode yang paling populer untuk pengendalian tingkat penemuan salah (FDR) berdasarkan nilai- p $\{p_i\}_{i=1}^m$ mungkin adalah prosedur Benjamini–Hochberg yang juga diusulkan pada tahun 1995 sebagai berikut:

- Definisikan nilai kritis $q_k = \frac{k}{m} \alpha$
- Urutkan nilai- p sebagai $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(m)}$
- Tetapkan $r = \max \left\{ i: p_{(i)} \leq \frac{i}{m} \alpha \right\}$. Jika r tidak ada, jangan lakukan penolakan; jika ada, tolak H_{i0} yang memenuhi $p_i \leq \frac{r}{m} \alpha$

Untuk nilai- p super-seragam, FDR dari prosedur BH memenuhi:

$$\alpha(\mathcal{F}) \leq \frac{m_0}{m} \alpha. \quad (10.72)$$

Hal ini memotivasi pengembangan “prosedur FDR adaptif” yang menggunakan estimasi $\pi_0 = \frac{m_0}{m}$ untuk menghabiskan tingkat FDR nominal yang telah ditentukan sebelumnya α . Pada intinya, dengan estimasi $\hat{\pi}_0$ yang baik dari π_0 , prosedur adaptif menggantikan nilai kritis $\left\{ \frac{i\alpha}{m} \right\}_{i=1}^m$ dalam prosedur BH dengan $\left\{ \frac{i\alpha}{m\hat{\pi}_0} \right\}_{i=1}^m$ dan dapat lebih ampuh daripada rekan-rekan non-adaptifnya. Ini didefinisikan sebagai berikut: misalkan



$$t_\alpha = \sup \left\{ t \in [0,1], \frac{\hat{\pi}_0(\lambda)t}{m^{-1}(R(t) \vee 1)} \leq \alpha \right\}$$

dan menolak yang mana $p_i \leq t_\alpha$.

Di sisi lain, perilaku FDR di bawah ketergantungan telah menarik banyak perhatian pengembangan tetapi belum sepenuhnya terselesaikan. Isu kunci untuk kontrol FDR di bawah ketergantungan adalah variabilitas variabel acak $FDP = \frac{V}{RV1}$, dan biasanya sulit untuk FDP menurunkan ketidaksetaraan konsentrasi untuk di bawah ketergantungan. Akhirnya, dalam studi klinis, genetika dan genomika nilai- p sering memiliki distribusi diskrit. Hal ini membuat prosedur yang dikembangkan untuk nilai- p kontinu menjadi kurang ampuh. Konjektur Simes dinyatakan sebagai berikut. Misalkan $\{p_i\}_{i=1}^m$ bergantung sedemikian rupa sehingga masing-masing p_i terdistribusi secara seragam pada, yaitu, $p_i \sim \text{Unif}[0,1]$. Definisikan $\bar{F}_m(t) = m^{-1} \sum_{i=1}^m \mathbf{1}_{\{p_i \leq t\}}$. Maka $\mathbb{E}[\bar{F}_m(t)] = t$. Ketidaksetaraan Simes adalah:

$$\Pr \left(\bigcap_{t \in [0, \alpha]} \left\{ \bar{F}_m(t) \leq \frac{t}{\alpha} \right\} \right) \geq 1 - \alpha \quad \text{for a fixed } \alpha \in (0,1). \quad (10.73)$$

Mengidentifikasi semua kondisi di mana ketidaksetaraan Persamaan (10.73) di atas berlaku tetap menjadi masalah yang belum terselesaikan.

Di sini, sebuah grafik memvisualisasikan ketergantungan di antara sekumpulan variabel acak. Dalam setiap grafik tersebut, sebuah simpul mewakili variabel acak, dan sebuah sisi antara dua simpul memiliki beberapa jenis hubungan antara variabel acak yang bersesuaian. Untuk model medan acak Markov, tidak adanya sisi antara dua simpul berarti bahwa variabel acak yang bersesuaian secara bersyarat independen dengan variabel lain. Model grafis ini digunakan dalam berbagai domain, misalnya, jaringan gen, interaksi antar spesies, dan lain-lain.

Grafik Markov dan Model Grafis Gaussian

Model grafis menampilkan hubungan antarvariabel sebagai simpul dan sisi. Bentuk ini membantu pembaca melihat struktur ketergantungan secara lebih visual. Sebuah graf $\mathcal{G}$ terdiri dari dua himpunan: V himpunan simpul dan E himpunan sisi. Dua titik dikatakan bertetangga jika ada sisi yang menghubungkannya. Sebuah jalur adalah sekumpulan simpul yang dihubungkan oleh sisi-sisi, dan sebuah graf lengkap adalah graf yang semua simpulnya dihubungkan oleh sebuah sisi. Graf Markov adalah graf sedemikian sehingga:

$$\mathbf{1}_{\{X \sim Y\}} = 0 \Leftrightarrow X \perp Y \mid \text{rest.}$$

Untuk subgraf A, B , dan C , C dikatakan memisahkan A dan B jika setiap jalur antara A dan B berpotongan dengan sebuah simpul di C . Pemisah memecah graf menjadi bagian-bagian yang saling independen secara bersyarat. Sebuah klik adalah subgraf lengkap. Lihat contoh 17.2 dalam teks. Fungsi kepadatan probabilitas f pada graf Markov dapat direpresentasikan sebagai:



$$f(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(x_c),$$

di mana $\mathcal{C}$ himpunan klik maksimal dan fungsi positif ψ_c disebut "potensial klik".

Catatan 10.28

Graf lengkap tidak secara unik menentukan struktur ketergantungan orde tinggi dalam distribusi gabungan variabel; lihat contoh 17.3 di buku teks.

Model grafis Gaussian. Misalkan $X = (X_1, \dots, X_p) \sim \text{Normal}(\mu, \Sigma)$ dan $\theta = \Sigma^{-1} = (\theta_{ij})$. Maka,

$$\theta_{ij} = 0 \Leftrightarrow X_i \perp X_j \mid \text{rest.}$$

Misalkan kita memiliki N pengamatan $x_1, \dots, x_N$ dari $\text{Normal}(\mu, \Sigma)$. Biarkan

$$S = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T,$$

Jika graf sudah lengkap, maka setelah maksimisasi parsial terhadap μ , kita peroleh:

$$l(\theta) = \log \det \theta - \text{trace}(S\theta). \quad (10.74)$$

Fungsi tujuan ini dimaksimalkan pada $\Sigma = S$. Sekarang misalkan graf tersebut tidak lengkap, θ_{ij} yaitu beberapa di antaranya bernilai nol. Maka, kita peroleh:

$$l(\theta) = \log \det \theta - \text{trace}(S\theta) - \sum_{(j,k) \notin E} \gamma_{jk} \theta_{jk},$$

melalui metode pengali Lagrange. Gradien untuk memaksimalkan persamaan di atas adalah:

$$\theta^{-1} - S - \Gamma = 0,$$

di mana Γ adalah matriks parameter Lagrange dengan entri bukan nol untuk semua pasangan dengan tepi yang tidak ada. Masalah optimasi di atas dapat diselesaikan menggunakan struktur distribusi bersyarat yang melekat pada distribusi Gaussian. Model grafis Gaussian jarang bertujuan untuk memperkirakan entri θ bukan nol dari melalui fungsi tujuan:

$$\log \det \theta - \text{trace}(S\theta) - \lambda \|\theta\|_1,$$

yang persamaan gradiennya adalah



$$\Theta^{-1} - S - \lambda \text{sgn}(\Theta) = 0,$$

di mana mengacu pada subgradien.

Graf Tak Berarah untuk Variabel Diskrit

“Model Ising” dalam mekanika statistik dan “mesin Boltzmann” dalam literatur mesin. Nilai pada setiap simpul dapat diamati (“terlihat”) atau tidak diamati (“tersembunyi”). Pertimbangkan terlebih dahulu kasus di mana semua p simpul terlihat dengan pasangan tepi (j, k) yang dihitung dalam E . Model dapat ditulis sebagai:

$$p(X, \theta) = \exp \left(\sum_{(j,k) \in E} \theta_{jk} X_j X_k - \Phi(\theta) \right) \quad \text{for } X \in \mathcal{X},$$

di mana $\mathcal{X} = \{0,1\}^p$. Isu utama dengan model ini adalah kompleksitas komputasinya. Aproksimasi medan rata-rata dapat digunakan, yang terkait dengan inferensi Bayes variasional. Selain itu, sampel Gibbs dapat digunakan untuk mendapatkan solusi yang mendekati tepat. Sekarang pertimbangkan kasus di mana $X_{\mathcal{V}}$ terlihat tetapi $X_{\mathcal{H}}$ tersembunyi. Maka *log-likelihood* dari data yang diamati adalah:

$$\begin{aligned} l(\theta) &= \sum_{i=1}^N \log[\Pr_{\theta}(X_{\mathcal{V}} = x_{i\mathcal{V}})] \\ &= \sum_{i=1}^N \log \sum_{x_{\mathcal{H}} \in \mathcal{X}_{\mathcal{H}}} \exp \left(\sum_{(j,k) \in E} \theta_{jk} X_j X_k - \Phi(\theta) \right). \end{aligned}$$

Graf Acak Eksponensial

Grafik acak eksponensial (ERG) diatur oleh distribusi probabilitas yang termasuk dalam keluarga eksponensial, sedemikian rupa sehingga struktur kombinatorial suatu graf sepenuhnya ditentukan oleh anggota keluarga tersebut. Secara matematis, model untuk ERG tak berarah dapat ditulis sebagai:

$$p_{\beta}(G) = \exp \left(\sum_{i=1}^k \beta_i T_i(G) - \psi(\beta) \right),$$

di mana $\beta = (\beta_1, \dots, \beta_k)$ adalah vektor parameter riil, $T_1, T_2, \dots, T_k$ adalah fungsi pada ruang graf (misalnya, jumlah sisi, segitiga, bintang, siklus, ...), dan ψ adalah konstanta normalisasi. Dalam hal ukuran relatif jumlah n sisi dan jumlah v simpul, graf dapat secara kasar diibaratkan sebagai dikategorikan menjadi yang jarang atau yang padat.

Catatan 10.29



Berikut adalah beberapa pertanyaan sulit yang terkait dengan ERG: transisi fase dalam ERG yang tersirat oleh ψ ; metrik pada graf jarang; spektrum dari graf jarang. Fungsi normalisasi ψ sangat sulit dihitung secara eksplisit, dan memperkirakan ERG menggunakan metode kemungkinan maksimum atau Bayesian biasanya sulit.

Statistik Eigen dari Matriks Kovarians Sampel

Diberikan pengamatan $x_1, \dots, x_N \in \mathbb{R}^p$ dari vektor acak $X \in \mathbb{R}^p$, matriks kovarians sampelnya adalah:

$$C = \frac{1}{N} \sum_{i=1}^N x_i x_i^T.$$

Matriks kovarians memainkan peran penting dalam analisis multivariat. Misalkan matriks kovarians dari X adalah Σ . Pertanyaan mendasar adalah “bagaimana dekomposisi spektral dari C berhubungan dengan dekomposisi spektral dari Σ ?” Pertanyaan ini berkaitan dengan pengujian struktur ketergantungan antara entri dari X , PCA berdasarkan C , dan lainnya. Misalkan $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_p$ adalah nilai eigen dari C .

$$\mu_p(t) = \frac{1}{p} \sum_{i=1}^p \mathbf{1}_{\{\lambda_i \leq t\}}.$$

Hukum Marchenko–Pastur (atau Hukum Seperempat Lingkaran)

Jika X memiliki entri yang saling bebas dan terdistribusi secara identik dengan nilai rata-rata 0 dan varians 1, $\min(N, p) \rightarrow \infty$ dan $Np^{-1} \rightarrow \rho \in (0, \infty)$, maka $\mu_p \rightarrow \mu$ dalam topologi-*lemah dalam distribusi, di mana

$$\mu(A) = \begin{cases} (1 - \rho^{-1}) \mathbf{1}_A(0) + \nu(A), & \text{if } \lambda \geq 1, \\ \nu(A), & \text{if } \lambda \in [0, 1]. \end{cases}$$

dan

$$\nu(dx) = \frac{1}{2\pi} \frac{\sqrt{((\rho_+ - x)(x - \rho_-))}}{\rho x} \mathbf{1}_{[\rho_-, \rho_+]} dx$$

dan $\rho_{\pm} = (1 \pm \sqrt{\rho})^2$.

Universalitas Tepi: Hukum Tracy–Widom

Dengan asumsi yang sama seperti sebelumnya, jika entri X memiliki ekor distribusi yang cukup tipis, maka nilai λ_1 eigen terbesar, setelah pemusatan dan penskalaan yang sesuai, secara asimtotik mengikuti hukum Tracy–Widom Tipe 1.



10.9 LATIHAN

1. Regresi

- Diberikan sebuah himpunan data harga rumah beserta fitur-fiturnya (ukuran, lokasi, jumlah kamar tidur, dan lain-lain), terapkan sebuah model regresi linear untuk memprediksi harga rumah baru.
- Jelaskan konsep penyesuaian berlebih (terlalu pas dengan data latih) dalam konteks regresi polinomial. Bagaimana Anda dapat mengatasi masalah ini menggunakan teknik-teknik seperti regularisasi (L1 dan L2)?
- Bandingkan dan kontraskan kinerja dari regresi linear, regresi panggung, dan regresi laso pada himpunan data yang diberikan. Analisis efek dari berbagai parameter regularisasi yang berbeda terhadap koefisien model.

2. Klasifikasi

- Terapkan model regresi logistik untuk mengklasifikasikan pesan surat elektronik sebagai pesan sampah atau bukan pesan sampah.
- Jelaskan batas keputusan dari mesin vektor pendukung dan bagaimana perbedaannya dengan regresi logistik.
- Bandingkan kinerja dari regresi logistik, mesin vektor pendukung, dan pengklasifikasi pohon keputusan pada himpunan data yang diberikan. Analisis efek dari berbagai fungsi inti yang berbeda (linear, Fungsi Basis Radial [FBR]) terhadap kinerja mesin vektor pendukung tersebut.

3. Kompleksitas Model

- Jelaskan kompromi antara simpangan dan ragam dalam konteks pembelajaran mesin. Bagaimana kompleksitas model (misalnya, jumlah fitur, parameter model) memengaruhi simpangan dan ragam?
- Uraikan teknik-teknik untuk pemilihan model, seperti validasi silang (lipatan- k , tinggalkan-satu-keluar), serta perannya dalam menemukan kompleksitas model yang paling optimal.
- Diskusikan pentingnya rekayasa fitur dalam meningkatkan kinerja model. Berikan contoh-contoh teknik rekayasa fitur (misalnya, penskalaan fitur, penyandian satu-kali, pengurangan dimensi).

4. Jaringan Saraf Tiruan

- Terapkan jaringan saraf tiruan umpan-maju sederhana dengan satu lapisan tersembunyi untuk mengklasifikasikan angka tulisan tangan (himpunan data MNIST).
- Jelaskan algoritma perambatan balik dan perannya dalam melatih jaringan saraf tiruan.
- Diskusikan konsep penyesuaian berlebih dalam pembelajaran mendalam serta teknik-teknik untuk regularisasinya, seperti pengguguran dan penghentian dini.

5. Mesin Vektor Pendukung

- Terapkan mesin vektor pendukung dengan fungsi inti linear untuk mengklasifikasikan himpunan data biner (dua kelas).



- b. Jelaskan konsep trik fungsi inti dan bagaimana hal tersebut memungkinkan mesin vektor pendukung untuk menangani data yang tidak dapat dipisahkan secara linear.
- c. Bandingkan kinerja mesin vektor pendukung dengan berbagai fungsi inti (linear, FBR, polinomial) pada himpunan data yang diberikan.

6. Algoritma Bayes

- a. Jelaskan konsep penarikan kesimpulan Bayes dan penerapannya dalam pembelajaran mesin.
- b. Terapkan pengklasifikasi Bayes Naif untuk klasifikasi teks (misalnya, analisis sentimen).
- c. Diskusikan kelebihan dan kekurangan dari metode Bayes dibandingkan dengan pendekatan frekuentis (berdasarkan frekuensi kejadian).

7. Model Grafis

- a. Jelaskan konsep jaringan Bayes dan rantai Markov.
- b. Nyatakan sebuah skenario yang diberikan (misalnya, sebuah sistem dari komponen-komponen yang saling berinteraksi) menggunakan jaringan Bayes.
- c. Diskusikan penerapan model grafis dalam berbagai bidang, seperti pemrosesan bahasa alami dan visi komputer.



BAB 11

ALGORITMA KRIPTOGRAFI YANG AMAN

11.1 KRIPTOGRAFI HEMAT DAYA

Bab ini menyajikan prinsip-prinsip dasar dan aplikasi algoritma kriptografi yang digunakan dalam komunikasi yang aman. Pembahasan dimulai dengan fungsi hash kriptografi dan perannya dalam memastikan integritas data. Kasus penggunaan utama seperti algoritma MD5, protokol Secure Sockets Layer (SSL), dan mekanisme hashing kata sandi dieksplorasi. Fokus tambahan diberikan pada desain algoritma aman untuk lingkungan berdaya rendah seperti perangkat Internet of Things (IoT). Aplikasi dalam teknologi blockchain dan tanda tangan digital disorot untuk menunjukkan implementasi praktis.

- Tujuan bab ini adalah: Untuk memperkenalkan algoritma kriptografi dan perannya dalam mencapai keamanan dan integritas data. Untuk menjelaskan struktur dan fungsionalitas fungsi hash kriptografi seperti MD5.
- Untuk mengeksplorasi penggunaan hashing dalam protokol seperti SSL dan dalam mekanisme perlindungan kata sandi.
- Untuk memeriksa pendekatan kriptografi ringan yang cocok untuk lingkungan terbatas atau berdaya rendah.
- Untuk menunjukkan aplikasi hashing dalam sistem blockchain dan tanda tangan digital.

Kriptografi memainkan peran penting di era modern, di mana jutaan pengguna terhubung melalui internet dan secara rutin bertukar informasi berharga dan sensitif. Sangat penting bagi organisasi - termasuk perusahaan, lembaga keuangan, dan lembaga pemerintah - untuk membangun saluran komunikasi yang aman di seluruh jaringan digital yang berkembang, sambil mempertahankan kinerja sistem yang optimal. Algoritma kriptografi memungkinkan tujuan keamanan utama seperti kerahasiaan, integritas, otentikasi, dan non-penolakan. Algoritma ini banyak digunakan untuk mengatasi tantangan yang berkaitan dengan perlindungan data, memastikan bahwa informasi tetap rahasia, tahan terhadap perubahan, dan hanya dapat diakses oleh pengguna yang berwenang.

Biasanya, teknik enkripsi mengubah teks biasa yang dapat dibaca menjadi teks sandi menggunakan fungsi matematika, dan data asli dapat dipulihkan melalui dekripsi menggunakan algoritma yang sesuai. Fungsi hash, juga dikenal sebagai ringkasan pesan atau fungsi satu arah, berbeda dari metode enkripsi konvensional karena tidak menggunakan kunci. Sebaliknya, fungsi hash menghasilkan output dengan panjang tetap dari data input, terutama untuk memverifikasi integritas pesan. Nilai hash yang dihasilkan berfungsi sebagai sidik jari digital dari pesan, memungkinkan verifikasi bahwa data belum diubah oleh entitas yang tidak berwenang, malware, atau bentuk gangguan lainnya.

Kriptografi melindungi informasi melalui aturan matematika. Tujuannya bukan hanya menyembunyikan pesan, tetapi juga memastikan identitas, integritas, dan kepercayaan dalam komunikasi digital. Dengan semakin banyaknya perangkat komunikasi yang ringkas dan perluasan IoT, terdapat peningkatan permintaan untuk tugas komputasi, termasuk operasi



kriptografi, yang dilakukan dengan menggunakan daya minimal. Misalnya, sangat penting bahwa implan medis berfungsi dengan aman dan eksklusif untuk pengguna yang ditunjuk, sambil mengonsumsi energi yang sangat rendah. Meskipun beberapa skema kriptografi standar ada untuk sistem yang tidak dibatasi daya, mengadaptasinya untuk perangkat berdaya sangat rendah sering kali memerlukan desain khusus yang ringan. Adaptasi ini melibatkan kompromi - terutama, pengurangan ketahanan keamanan untuk mengakomodasi penggunaan daya yang lebih rendah. Akibatnya, algoritma kriptografi untuk lingkungan yang terbatas dioptimalkan dengan cermat untuk menyeimbangkan keamanan dengan efisiensi energi.

11.2 KRIPTOGRAFI KUNCI RAHASIA

Kriptografi kunci simetris tetap menjadi pendekatan dasar untuk memastikan komunikasi yang aman. Advanced Encryption Standard (AES) adalah standar yang berlaku untuk enkripsi simetris. Namun, bahkan konfigurasi yang paling ringan pun dapat membutuhkan komputasi yang intensif untuk perangkat dengan keterbatasan sumber daya yang parah. Untuk mengatasi hal ini, dua algoritma sandi simetris alternatif telah dikembangkan: algoritma Simon dan Speck. Simon dioptimalkan untuk kinerja yang efisien di lingkungan perangkat keras, sedangkan Speck dirancang untuk implementasi berbasis perangkat lunak. Meskipun sandi ringan ini beroperasi secara efektif pada perangkat keras minimal, mereka mengorbankan throughput - kompromi yang dapat diterima mengingat pertukaran data yang biasanya terbatas pada perangkat berdaya rendah. Selain itu, algoritma ini menawarkan margin keamanan yang lebih rendah dibandingkan dengan AES. Namun demikian, pertukaran ini sering kali dapat dibenarkan. Pertama, pihak lawan kemungkinan akan menemukan lalu lintas komunikasi yang jauh lebih sedikit untuk dicegat atau dianalisis. Kedua, banyak perangkat berdaya rendah tidak memiliki perlindungan fisik, sehingga lebih rentan terhadap serangan berbasis perangkat keras - seperti eksploitasi saluran samping - daripada metode kriptanalitik yang kompleks.

11.3 KRIPTOGRAFI KUNCI PUBLIK

Kriptografi kunci publik, meskipun mendasar untuk pertukaran kunci yang aman, secara tradisional membutuhkan sumber daya komputasi yang tinggi dan umumnya dianggap tidak cocok untuk tugas enkripsi rutin di lingkungan dengan sumber daya terbatas. Akibatnya, kriptografi ini terutama digunakan untuk memfasilitasi distribusi kunci yang aman untuk sistem enkripsi simetris. Dengan meningkatnya adopsi kerahasiaan maju sempurna di seluruh protokol komunikasi modern dan meningkatnya ketergantungan pada perangkat seluler dan IoT, permintaan akan kriptografi kunci publik yang hemat energi telah meningkat pesat. Sebagian besar skema kunci publik bergantung pada aritmatika modular, termasuk algoritma seperti Rivest–Shamir–Adleman (RSA), Diffie–Hellman (DH), dan sistem pendukung keputusan. Untuk mempercepat komputasi, teknik seperti Teorema Sisa Cina sering digunakan; namun, biaya komputasi tetap signifikan, terutama ketika ukuran kunci meningkat.

Kemajuan terbaru dalam riset kriptografi telah menggarisbawahi potensi kriptografi kurva eliptik (ECC) sebagai alternatif yang aman dan efisien untuk sistem berdaya rendah. ECC

The diagram illustrates a neural network architecture for sequence processing. The input sequence is A, B, C, D, B. The output sequence is A, B, C, D, B. The network consists of several layers: an input layer, a hidden layer with nodes A, B, C, D, B, and an output layer. The hidden layer nodes are connected to the input and output nodes via various paths, including direct connections and paths through intermediate nodes like F and W_t , K_t .

Protokol Pertukaran Kunci

Diffie dan Hellman mengembangkan gagasan ini lebih jauh dengan mengusulkan penggunaan *fungsi pintu jebakan*—sebuah fungsi yang mudah dihitung dalam satu arah namun sulit untuk dibalikkan tanpa adanya pengetahuan khusus. Mereka membuktikan bahwa fungsi semacam itu dapat memungkinkan beroperasinya *penyandian kunci publik* maupun *tanda tangan digital*, sehingga mendukung komunikasi yang aman dan dapat dibuktikan kebenarannya melalui saluran terbuka. Meskipun perpangkatan bersisa (modular) disarankan oleh John Gill sebagai calon yang berpotensi untuk fungsi-fungsi tersebut, kemampuannya untuk dibalikkan sebagai sebuah pintu jebakan masih belum pasti. Namun demikian, hal ini memungkinkan terciptanya ragam yang aman dari tata cara Merkle—yang kini dikenal sebagai pertukaran kunci DH. Mekanisme ini juga meletakkan landasan bagi penyandian ElGamal, meskipun kebutuhan akan penyandian berpeluang (probabilistik) untuk tujuan keamanan tidak langsung disadari pada saat itu.



Menyusul terbitan penting mereka pada tahun 1976, pencarian terhadap fungsi pintu jebakan yang praktis semakin gencar. Pada tahun 1977, Rivest, Shamir, dan Adleman memperkenalkan algoritma RSA, sebuah rancangan ilmu sandi yang digunakan secara luas berdasarkan pada sulitnya pemfaktoran bilangan bulat. RSA, bersama dengan metode-metode berbasis DH, terus merajai penerapan penyandian kunci publik dan tanda tangan digital. Pada tahun 1978, Rabin mengusulkan sebuah fungsi pintu jebakan pengganti yang pembalikannya terbukti sama sulitnya dengan pemfaktoran. Namun, baik fungsi RSA maupun Rabin harus diwujudkan secara hati-hati dengan penambahan data pengisi serta susunan yang tepat untuk memastikan keamanan CPA atau CCA, karena penerapan yang asal-asalan dapat menjadi rentan. Penelusuran selanjutnya dilakukan oleh Lamport, Goldwasser, Micali, Rivest, dan lainnya mengungkapkan bahwa tanda tangan digital dapat dibangun dari fungsi satu arah mana pun. Hal ini menunjukkan bahwa fungsi pintu jebakan, meskipun berguna, tidak secara mutlak diperlukan untuk semua unsur dasar ilmu sandi.

Fungsi Pintu Jebakan

Fungsi pintu jebakan (juga disebut permutasi pintu jebakan - TDP) adalah kelas khusus fungsi satu arah yang penting untuk membangun sistem kriptografi kunci publik. Tidak seperti fungsi satu arah atau permutasi pseudorandom (PRP) - yang dengan sendirinya tidak secara inheren mendukung enkripsi kunci publik - fungsi pintu jebakan mencakup fitur struktural yang penting: fungsi tersebut didefinisikan sebagai koleksi fungsi berkunci dengan kunci terpisah untuk komputasi dan inversi. Setiap fungsi pintu jebakan memiliki dua kunci terkait: kunci publik yang digunakan untuk menghitung fungsi ke arah depan dan kunci privat ("jebakan") yang digunakan untuk inversi yang efisien. Bahkan jika kunci publik sepenuhnya diungkapkan kepada pihak lawan, membalikkan fungsi tetap tidak mungkin dilakukan secara komputasi tanpa jebakan. Hal ini menjadikan fungsi pintu jebakan sebagai blok bangunan fundamental untuk skema enkripsi dan tanda tangan yang aman dalam kriptografi kunci publik.

Definisi 11.1 (Fungsi Pintu Jebakan)

Sebuah *kumpulan fungsi pintu jebakan* mengacu pada sebuah himpunan $\mathcal{F}$ yang berisi fungsi-fungsi berhingga, di mana setiap fungsi $f \in \mathcal{F}$ adalah pemetaan satu-satu dari sebuah daerah asal S_f ke sebuah daerah hasil T_f . Kumpulan semacam itu harus memenuhi sifat-sifat berikut:

- **Pembuatan, Perhitungan, dan Pembalikan yang Efisien:** Terdapat sebuah algoritma waktu-polinomial berpeluang G yang, jika diberikan masukan 1^n , menghasilkan sebuah pasangan (f, f^{-1}) . Ini merupakan bentuk perwakilan yang ringkas (dengan ukuran polinomial dalam n) dari fungsi maju dan fungsi balikan. Perhitungan nilai $f(x)$ dan $f^{-1}(y)$ harus dapat dihitung secara efisien dalam waktu polinomial.
- **Pengambilan Sampel yang Efisien:** Terdapat sebuah algoritma waktu-polinomial berpeluang yang, ketika diberikan sebuah fungsi f , dapat mengambil sampel sebuah unsur acak dari S_f , atau dari sebuah sebaran yang secara statistik mendekati seragam pada S_f .



- **Kesatu-arahan:** Fungsi f secara perhitungan sulit untuk dibalik. Secara khusus, untuk setiap pihak lawan waktu-polinomial berpeluang A , terdapat sebuah fungsi yang dapat diabaikan $\epsilon(n)$ sedemikian rupa sehingga:

$$\Pr_{(f, f^{-1}) \leftarrow G(1^n), x \leftarrow S_f} [A(1^n, f, f(x)) = x] < \epsilon(n).$$

Perlu dicatat bahwa definisi ini sedikit menyimpang dari definisi yang diajukan oleh Boneh dan Shoup, yang membatasi domain S agar independen dari fungsi f . Meskipun demikian, *fungsi pintu jebakan* seperti RSA dan Rabin dapat dirumuskan ulang agar sesuai dengan varian definisi yang lebih ketat, bahkan memungkinkan $S_f = \{0,1\}^n$. Dalam kriptografi teoretis, sering diasumsikan bahwa terdapat keluarga TDP dengan domain dan rentang yang sama dengan $\{0,1\}^n$. Asumsi ini umumnya disebut sebagai **Aksioma TDP**. Struktur formal yang sesuai kadang-kadang disebut sebagai koleksi TDP yang ditingkatkan ganda;

Fungsi Pintu Jebakan Rabin

Masalah matematika mengenai bilangan prima dan pemfaktoran bilangan bulat telah menjadi bahan riset mendalam selama berabad-abad. Terlepas dari pengkajian yang meluas, belum ada algoritma yang diketahui mampu memfaktorkan bilangan bulat dengan n -angka dalam waktu polinomial. Hal ini mengarah pada keyakinan yang dipegang luas bahwa algoritma pemfaktoran yang mangkus semacam itu memang tidak ada. Untuk penerapan ilmu sandi, sebuah pengandaian yang lebih kuat biasanya diperlukan—yakni, bahwa memfaktorkan bilangan bulat Blum acak adalah suatu hal yang sulit secara perhitungan. Sebuah bilangan bulat Blum didefinisikan sebagai hasil kali $n = pq$, di mana baik p maupun q adalah bilangan prima yang kongruen dengan $3 \pmod{4}$. Misalkan $\mathcal{B}_n$ menyatakan himpunan bilangan prima semacam itu:

$$\mathcal{B}_n = \{P \in [1, 2^n] \mid P \text{ is prime and } P \equiv 3 \pmod{4}\}.$$

Aksioma Pemfaktoran

Untuk setiap algoritma polinomial-waktu probabilistik A , terdapat fungsi yang dapat diabaikan $\epsilon(n)$ sedemikian sehingga:

$$\Pr_{P, Q \leftarrow \mathcal{B}_n} [A(P \cdot Q) = \{P, Q\}] \leq \epsilon(n).$$

Berdasarkan asumsi ini, konstruksi berikut menggambarkan fungsi pintu jebakan Rabin (dalam versi yang disempurnakan oleh Blum dan Williams):

- **Pembangkitan Kunci:** Pilih dua bilangan prima acak P dan Q dengan panjang n sehingga $P, Q \equiv 3 \pmod{4}$, lalu hitung $N = P \cdot Q$. Perhatikan bahwa $\phi(N) \equiv 0 \pmod{4}$ karena $(P-1)(Q-1)$ habis dibagi 4.
- **Kunci Publik:** N .
- **Kunci Pribadi (Pintu Jebakan):** Faktor-faktor P dan Q .
- **Evaluasi Maju:** Definisikan fungsi Rabin sebagai:



$$\text{RABIN}_N(X) = X^2 \bmod N.$$

Fungsi ini memetakan elemen dari $\mathbb{Z}_N^*$ ke residu kuadratnya modulo N .

- **Prosedur Inversi:** Diberikan $Y = X^2 \bmod N$, tujuannya adalah untuk memulihkan X .
Hitung:

$$A = Y \bmod P, \quad B = Y \bmod Q.$$

Karena $P = 4t + 3$ dan $Q = 4t' + 3$, hitung:

$$X_1 = A^{t+1} \bmod P, \quad X_2 = B^{t'+1} \bmod Q.$$

Ini menghasilkan akar kuadrat dari Y modulo P dan Q , masing-masing. Gunakan Teorema Sisa Cina untuk menggabungkan X_1 dan X_2 dan memperoleh $X' \in \mathbb{Z}_N^*$. Kebenaran metode ini bergantung pada fakta bahwa jika X sendiri merupakan residu kuadratik, yaitu $X = S^2 \bmod N$, maka invers yang dihitung akan sesuai dengan X . Hal ini dipastikan karena:

$$X_1 \equiv S^2 \bmod P, \quad X_2 \equiv S^2 \bmod Q \Rightarrow X' \equiv X \bmod N$$

Perlu juga dicatat bahwa pengambilan sampel dari distribusi yang mendekati seragam pada himpunan residu kuadrat modulo N yang dilambangkan QR_N dapat dicapai dengan memilih bilangan bulat acak $S \in \{1, \dots, N-1\}$ dan menghitung $X = S^2 \bmod N$.

Sifat Satu Arah dari Fungsi Rabin

Keamanan fungsi Rabin bergantung pada sifat satu arahnya, yang dapat dibuktikan menggunakan lemma berikut:

Lemma 11.1.

Misalkan X dan Y bilangan bulat sedemikian sehingga $X \not\equiv \pm Y \pmod{N}$, tetapi $X^2 \equiv Y^2 \pmod{N}$. Maka:

$$\gcd(X - Y, N) \notin \{1, N\}.$$

Bukti:

Asumsikan $X \not\equiv \pm Y \pmod{N}$, yang menyiratkan bahwa baik $X - Y$ maupun $X + Y$ tidak habis dibagi N . Karena $X^2 \equiv Y^2 \pmod{N}$, diperoleh:

$$X^2 - Y^2 = (X - Y)(X + Y) \equiv 0 \pmod{N} \Rightarrow N \mid (X - Y)(X + Y)$$

Hal ini menyiratkan bahwa $\gcd(X - Y, N)$ harus merupakan faktor nontrivial dari N , yaitu bukan 1 maupun N . Lemma ini mengarah pada implikasi penting: diberikan pasangan (X, Y)



dengan $X \not\equiv \pm Y \pmod{N}$ dan $X^2 \equiv Y^2 \pmod{N}$, seseorang dapat menghitung $\gcd(X - Y, N)$ untuk memperoleh faktor nontrivial dari N . Untuk $N = PQ$, hasil ini mengungkapkan P atau Q , dan faktor lainnya dapat ditemukan dengan pembagian sederhana.

Selanjutnya, pertimbangkan lawan yang memiliki akses ke algoritma A yang membalikkan fungsi Rabin dengan probabilitas keberhasilan ϵ . Jika input acak X dipilih dan $Y = A(X^2)$ dihitung, maka dengan probabilitas setidaknya $\epsilon/2$, nilai Y yang dikembalikan memenuhi $Y^2 \equiv X^2 \pmod{N}$ dan $Y \not\equiv \pm X \pmod{N}$. Dalam kasus tersebut, memfaktorkan N menjadi mungkin. Hal ini menegaskan bahwa fungsi Rabin merupakan fungsi pintu jebakan berdasarkan asumsi sulitnya faktorisasi bilangan bulat.

Fungsi RSA

Fungsi RSA, yang diperkenalkan pada tahun 1977 oleh Rivest, Shamir, dan Adleman, adalah contoh TDP pertama yang diadopsi secara luas dan tetap menjadi salah satu konstruksi kriptografi yang paling banyak digunakan hingga saat ini.

- **Pembuatan Kunci:** Pilih dua bilangan prima acak besar P dan Q , masing-masing dengan panjang ℓ , lalu hitung modulus $N = P \cdot Q$. Fungsi totien Euler diberikan oleh $\varphi(N) = (P - 1)(Q - 1)$, yang sama dengan orde grup perkalian $\mathbb{Z}_N^*$. Pilih eksponen publik e sedemikian sehingga $e \in \mathbb{Z}_{\varphi(N)}^*$, yaitu $\gcd(e, \varphi(N)) = 1$.
- **Kunci Publik (Kunci Maju):** Pasangan (N, e) .
- **Kunci Pribadi (Kunci Pintu Jebakan):** Eksponen dekripsi d yang memenuhi $d \equiv e^{-1} \pmod{\varphi(N)}$. Dengan kata lain, $ed = k\varphi(N) + 1$ untuk suatu bilangan bulat k . Nilai d dapat dihitung secara efisien menggunakan Algoritma Euclidean yang Diperluas, asalkan faktorisasi N , yaitu P dan Q , diketahui.
- **Enkripsi (Evaluasi Maju):** Diberikan pesan $X \in \mathbb{Z}_N^*$, fungsi enkripsi RSA didefinisikan sebagai:

$$\text{RSA}_{N,e}(X) = X^e \pmod{N}.$$

- **Dekripsi (Inversi):** Fungsi RSA merupakan permutasi atas $\mathbb{Z}_N^*$ dan dapat diinversi menggunakan eksponen privat d . Dengan diberikan $Y = X^e \pmod{N}$, pesan asli dipulihkan sebagai:

$$X = Y^d \pmod{N}.$$

Hal ini mengikuti teorema Euler, yang menyatakan bahwa untuk setiap $X \in \mathbb{Z}_N^*$, berlaku $X^{\varphi(N)} \equiv 1 \pmod{N}$. Oleh karena itu:

$$X^{ed} = X^{k\varphi(N)+1} = X^{k\varphi(N)} \cdot X \equiv 1^k \cdot X \equiv X \pmod{N}.$$

Penting juga dicatat bahwa pengambilan sampel elemen acak dari $\mathbb{Z}_N^*$ dapat dilakukan dengan memilih bilangan bulat $X \in \{0, 1, \dots, N - 1\}$ dan memverifikasi bahwa $\gcd(X, N) = 1$. Metode ini berhasil dengan probabilitas tinggi karena $\mathbb{Z}_N^*$ mengandung $(P - 1)(Q - 1)$ elemen, yang



mencakup hampir semua bilangan bulat modulo N kecuali sebagian kecil yang dapat diabaikan.

Asumsi RSA

Asumsi RSA menyatakan bahwa fungsi RSA berperilaku sebagai TDP. Asumsi ini dianggap lebih kuat daripada asumsi yang diterima secara luas bahwa memfaktorkan hasil perkalian dua bilangan prima besar yang dipilih secara acak adalah sulit secara komputasi. Namun, masih menjadi masalah terbuka apakah kedua asumsi ini setara. Secara khusus, saat ini tidak diketahui apakah algoritma efisien dapat membalikkan fungsi RSA tanpa juga mampu memfaktorkan modulus N . Oleh karena itu, keamanan RSA didasarkan pada asumsi yang mungkin lebih kuat daripada kesulitan faktorisasi bilangan bulat.

Menggunakan Fungsi Pintu Jebakan untuk Enkripsi Kunci Publik: *Latihan:* Rancang skema enkripsi kunci publik berdasarkan TDP. Buktikan kebenarannya dan analisis keamanannya di bawah model serangan teks terang yang dipilih dan teks sandi yang dipilih.

Pertukaran Kunci dan Protokol DH: Salah satu pendekatan standar untuk memungkinkan komunikasi aman antara dua pihak, seperti Alice dan Bob, melalui saluran yang berpotensi tidak aman dan dipantau oleh penyadap, Eve, adalah menggunakan sistem enkripsi kunci publik. Dalam pengaturan ini, Alice menghasilkan pasangan kunci (e, d) , memublikasikan kunci publik e , dan Bob mengenkripsi pesan untuk Alice menggunakan kunci tersebut.

Namun, ini bukan satu-satunya metode untuk membangun rahasia bersama. Pendekatan alternatif dan banyak digunakan melibatkan penggunaan *protokol pertukaran kunci*. Protokol pertama - yang masih fundamental hingga saat ini - diperkenalkan oleh Diffie dan Hellman dalam karya perintis mereka tentang kriptografi kunci publik. Protokol ini dibangun berdasarkan sifat-sifat matematis dari grup siklik. Misalkan $\mathbb{Z}_P^*$ menyatakan grup perkalian bilangan bulat modulo bilangan prima P . Grup ini bersifat siklik, artinya terdapat generator $g \in \mathbb{Z}_P^*$ sedemikian sehingga:

$$\mathbb{Z}_P^* = \{g^0, g^1, g^2, \dots, g^{P-2}\} \bmod P.$$

Untuk setiap elemen $X \in \mathbb{Z}_P^*$, terdapat bilangan bulat sedemikian $i \in \{0, \dots, P-2\}$ sehingga $X \equiv g^i \bmod P$. Eksponen i ini disebut sebagai *logaritma diskrit* dari X terhadap g . Meskipun secara komputasi efisien untuk menemukan generator g untuk bilangan prima tertentu P , menghitung logaritma diskrit, yaitu menemukan i sedemikian sehingga $g^i \equiv X \bmod P$, diyakini secara komputasi tidak dapat dipecahkan. Ketidakmampuan inilah yang mendasari keamanan *protokol pertukaran kunci* DH. Jelas, jika Eve dapat menghitung logaritma diskrit untuk mengambil X dari $\hat{X}$ atau Y dari $\hat{Y}$, maka protokol DH menjadi tidak aman. Akibatnya, asumsi bahwa pertukaran kunci DH aman secara inheren menyiratkan kondisi yang lebih kuat daripada sekadar mengasumsikan bahwa masalah logaritma diskrit sulit dipecahkan (ekuivalen dengan menyatakan bahwa eksponensiasi merupakan permutasi satu arah—OWP).

Namun, sejauh yang diketahui saat ini, asumsi ini saja *tidak cukup* untuk memastikan keamanan protokol DH. Sebaliknya, kita memerlukan asumsi yang lebih kuat berikut. **Asumsi**



Decisional Diffie–Hellman (DDH) (Versi 1). Untuk setiap bilangan prima P dan generator g dari $\mathbb{Z}_P^*$, dua distribusi berikut, A dan B , atas rangkap tiga secara komputasional tidak dapat dibedakan:

$$A = \langle g^X, g^Y, g^{XY} \rangle \quad \text{dan} \quad B = \langle g^X, g^Y, Z \rangle,$$

di mana X dan Y dipilih secara seragam dan independen dari $\{1, \dots, P-2\}$, sedangkan Z dipilih secara seragam dari $\mathbb{Z}_P^*$. Di bawah asumsi DDH, dari perspektif Eve, kunci bersama k tampak tidak dapat dibedakan dari elemen acak dari $\mathbb{Z}_P^*$, sehingga penggunaannya sebagai kunci dalam skema enkripsi kunci privat apa pun menjadi aman. Misalnya, Bob dapat mengirim pesan m sepanjang ℓ kepada Alice secara aman dengan mentransmisikan $k \oplus m$. Namun, asumsi DDH tidak berlaku (meskipun secara luas dianggap “benar secara moral”) pada grup $\mathbb{Z}_P^*$ karena alasan sederhana: diberikan suatu elemen $\hat{Y} \in \mathbb{Z}_P^*$, seseorang dapat secara efisien menentukan apakah elemen tersebut merupakan residu kuadrat modulo P . Telah diketahui bahwa g^X merupakan residu kuadrat jika dan hanya jika X genap. Oleh karena itu, diberikan g^X dan g^Y , seseorang dapat mendeteksi dengan probabilitas $1/4$ apakah X dan Y keduanya genap; dalam kasus ini, g^{XY} akan selalu menjadi residu kuadrat, sedangkan elemen yang dipilih secara acak di $\mathbb{Z}_P^*$ merupakan residu kuadrat hanya dengan probabilitas $1/2$.

Untungnya, asumsi DDH diyakini berlaku untuk grup khusus lainnya. Salah satu contoh yang menonjol adalah subgrup residu kuadratik modulo P , khususnya ketika P dipilih sebagai $P = 2Q + 1$ untuk suatu bilangan prima Q . **Berbagai Jenis Permutasi:** Sangat penting untuk tidak mengacaukan konsep PRP, OWP, dan TDP. Baik PRP maupun OWP merupakan primitif simetris yang terkait dengan *kriptografi kunci privat*. OWP adalah fungsi tunggal (bukan kumpulan fungsi), mudah dihitung, tetapi secara komputasional tidak layak untuk dibalik. Sebaliknya, PRP merujuk pada kumpulan fungsi yang tampak tidak dapat dibedakan dari permutasi yang benar-benar acak bagi musuh *tanpa pengetahuan tentang kunci rahasia*. Perbedaan utama dengan TDP adalah bahwa bahkan ketika musuh mengetahui *forward key*, proses inversi tetap sulit secara komputasional kecuali musuh juga memiliki kunci balik (*trapdoor*) yang sesuai, yang membuat inversi menjadi mudah.

MD5

MD5 adalah algoritma hash kriptografi yang menerima masukan dengan panjang sembarang dan menghasilkan ringkasan pesan 128-bit. Algoritma ini mempertahankan keadaan internal 128-bit yang dibagi menjadi empat kata 32-bit, yang umumnya dilambangkan sebagai A , B , C , dan D . Pada awalnya, variabel-variabel keadaan ini diatur ke konstanta yang telah ditentukan. Algoritma memproses pesan masukan dalam blok-blok 512-bit berurutan; masing-masing blok melalui empat tahap (disebut *rounds*). Setiap *round* terdiri atas 16 operasi serupa yang melibatkan fungsi nonlinier, penjumlahan modular, dan rotasi kiri bitwise. MD5 menggunakan empat fungsi nonlinier yang berbeda—satu untuk setiap *round*—yang memanfaatkan operasi bitwise dasar: XOR ($\oplus$), AND ($\wedge$), OR ($\vee$), dan NOT ($\neg$). Fungsi-fungsi tersebut didefinisikan sebagai berikut:



$$F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$$

$$G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$$

$$H(B, C, D) = B \oplus C \oplus D$$

$$I(B, C, D) = C \oplus (B \vee \neg D)$$

Hashing Kata Sandi

Fungsi hash kriptografi menunjukkan properti satu arah, artinya secara komputasi tidak mungkin untuk membalikkan proses hashing untuk mengambil kembali masukan asli. Fungsi hash memetakan data masukan dengan ukuran apa pun ke "sidik jari" dengan panjang tetap, di mana bahkan sedikit perubahan pada masukan secara signifikan mengubah keluaran. Hash banyak digunakan untuk mengamankan kata sandi dengan memastikan bahwa meskipun penyimpanan kata sandi dikompromikan, kata sandi sebenarnya tetap terlindungi. Secara bersamaan, hash memungkinkan autentikasi dengan memverifikasi kecocokan kata sandi secara efisien. Alur kerja hashing kata sandi tipikal untuk pendaftaran dan autentikasi akun diberikan di bawah ini:

- Pengguna membuat akun dengan memilih kata sandi.
- Kata sandi yang dimasukkan di-hash, mengubahnya menjadi sidik jari dengan panjang tetap, dan disimpan dengan aman di dalam basis data.
- Ketika pengguna mencoba masuk kemudian, sistem melakukan hashing kata sandi yang diberikan dan membandingkannya dengan hash yang tersimpan.
- Jika hash cocok, pengguna diberikan akses; jika tidak, sistem memberi tahu pengguna bahwa kredensial yang diberikan tidak valid.
- Langkah 3 dan 4 diulangi setiap kali upaya login terjadi.

Catatan 11.1

Untuk meningkatkan keamanan, ketika autentikasi gagal, sistem (atau sistem operasi) harus menampilkan pesan kesalahan umum seperti "Nama pengguna atau kata sandi tidak valid," alih-alih secara eksplisit menyatakan kredensial mana yang salah. Praktik ini mencegah penyerang mengidentifikasi nama pengguna yang valid tanpa mengetahui kata sandi yang sesuai.

Catatan 11.2

Penting untuk membedakan antara fungsi hash kriptografi yang digunakan untuk hashing kata sandi dan fungsi hash yang digunakan dalam struktur data (misalnya, tabel hash). Fungsi hash yang digunakan dalam struktur data dioptimalkan untuk kecepatan, bukan keamanan. Sebaliknya, fungsi hash kriptografi, seperti SHA-256, SHA-512, RIPEMD, dan WHIRLPOOL, memprioritaskan keamanan dan ketahanan terhadap tabrakan, sehingga cocok untuk hashing kata sandi.

eskipun hashing kata sandi dengan fungsi hash kriptografi mungkin awalnya tampak cukup untuk keamanan, pendekatan ini saja tidak menjamin keamanan kata sandi. Penyerang dapat memulihkan kata sandi yang di-hash menggunakan berbagai metode sederhana, termasuk serangan kamus, terutama pada kata sandi yang sederhana atau mudah ditebak. Namun,



penerapan teknik keamanan tambahan—seperti salting dan key stretching—secara signifikan mengurangi efektivitas serangan tersebut.

Secure Sockets Layer

Tujuan utama protokol SSL adalah memastikan komunikasi yang aman melalui kanal jaringan. SSL beroperasi melalui arsitektur berlapis dan mendukung enkapsulasi berbagai protokol tingkat tinggi. Protokol Handshake SSL bertanggung jawab membangun autentikasi antara klien dan server. Selama fase handshake, kedua pihak menegosiasikan algoritma enkripsi dan kunci kriptografi sebelum data tingkat aplikasi apa pun dikirimkan. Salah satu manfaat utama SSL adalah independensi terhadap protokol aplikasi, sehingga SSL sesuai untuk mengamankan berbagai komunikasi jaringan. SSL memastikan keandalan dan integritas pesan dengan menggunakan *message authentication code* (MAC) berkunci, sebagaimana ditentukan dalam. Pustaka OpenSSL ssl menyediakan beberapa struktur data inti, sebagaimana dijelaskan dalam dokumentasi resmi:

- a. **SSL_METHOD (Metode SSL):** Mendefinisikan metode internal yang diimplementasikan dalam pustaka SSL untuk mendukung berbagai versi protokol (misalnya, SSLv1, SSLv2, TLSv1). Ini diperlukan untuk membuat objek SSL_CTX.
- b. **SSL_CIPHER (Cipher SSL):** Berisi informasi tentang cipher tertentu, yang merupakan bagian inti dari protokol SSL/Transport Layer Security (TLS). Rangkaian cipher dikonfigurasi pada tingkat SSL_CTX, dan cipher aktual yang digunakan selama sesi disimpan dalam struktur SSL_SESSION.
- c. **SSL_CTX (Konteks SSL):** Merepresentasikan konteks global yang digunakan oleh server atau klien sepanjang masa pakai aplikasi. Ini menyimpan nilai konfigurasi default untuk koneksi SSL dan sangat penting untuk membuat instance koneksi individual.
- d. **SSL_SESSION (Sesi SSL):** Mengkapsulasi detail sesi TLS/SSL aktif, termasuk cipher yang digunakan, sertifikat, dan kunci kriptografi yang terkait dengan server dan klien.
- e. **SSL (Koneksi SSL):** Struktur utama yang digunakan untuk mengelola koneksi SSL/TLS. Dibuat untuk setiap instance koneksi, struktur ini terhubung ke struktur relevan lainnya dan merupakan antarmuka utama yang berinteraksi dengan aplikasi selama waktu berjalan.

Blockchain

Konsep buku besar terdistribusi dan terdesentralisasi dapat dikatakan sebagai salah satu inovasi paling transformatif sejak munculnya World Wide Web. Selama bertahun-tahun, teknologi blockchain telah menemukan banyak aplikasi, dengan mata uang digital menjadi salah satu yang paling menonjol. Pada bagian ini, kita akan meneliti struktur data inti yang umum digunakan dalam mata uang kripto yang dibangun di atas teknologi blockchain. Blockchain pada dasarnya adalah struktur data berbasis penunjuk hash yang terdiri dari blok-blok yang terhubung. Setiap blok berisi komponen-komponen kunci berikut:

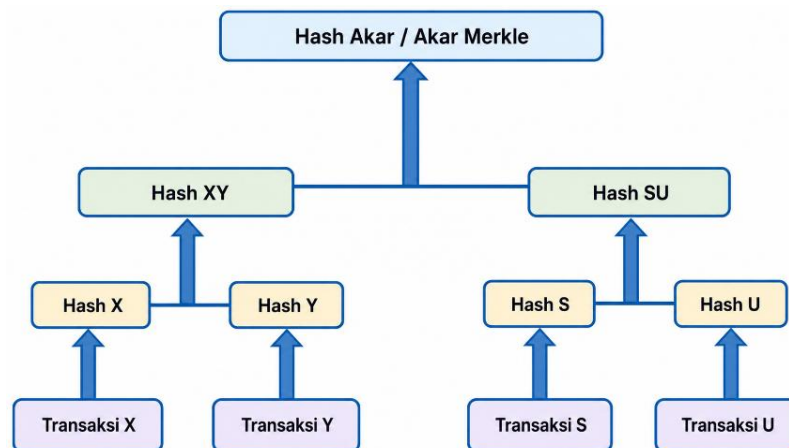
- a. **Indeks:** Menunjukkan posisi blok dalam rantai. Misalnya, blok pertama (blok genesis) memiliki indeks 0.



- b. **Hash:** Pengidentifikasi unik untuk blok, yang diperoleh dari hashing isi blok. Bitcoin menggunakan SHA-256, varian dari keluarga SHA-2, sedangkan Ethereum menggunakan SHA-3.
- c. **PreviousHash:** Hash dari blok sebelumnya, membentuk tautan kriptografi antar blok—oleh karena itu istilah "blockchain."
- d. **Timestamp:** Stempel waktu UTC Unix yang menunjukkan kapan blok dibuat.
- e. **Nonce:** Bilangan bulat 32- atau 64-bit yang digunakan dalam proses penambangan untuk memenuhi kondisi kesulitan.
- f. **numTx:** Mewakili jumlah transaksi yang terdapat dalam blok.
- g. **Transaksi:** Sebuah array yang berisi semua transaksi yang termasuk dalam blok.

Catatan 11.3

Untuk mengoptimalkan penyimpanan, beberapa implementasi menggunakan pohon Merkle (lihat Gambar 11.2) untuk merepresentasikan data transaksi secara efisien.



Gambar 11.2 Contoh pembangkitan hash akar Merkle dalam struktur blockchain.

Setiap blok dalam blockchain berisi beberapa transaksi valid yang di-hash dan distrukturkan menggunakan pohon Merkle. Hash dari blok sebelumnya disertakan dalam blok saat ini, menciptakan rantai kriptografi yang menghubungkan blok-blok tersebut. Penghubungan rekursif ini memastikan integritas rantai dari blok genesis dan seterusnya. Terkadang, dua blok dapat dihasilkan secara bersamaan, menghasilkan percabangan sementara. Setiap node dalam jaringan menerapkan algoritma konsensus untuk menentukan percabangan mana yang akan diperpanjang, biasanya memilih rantai dengan skor kumulatif tertinggi—seperti rantai dengan pekerjaan komputasi terbanyak. Blok yang tidak termasuk dalam rantai utama disebut sebagai blok *yatim piatu*.

Node yang berbeda mungkin mempertahankan versi blockchain yang sedikit berbeda. Namun, mereka terus memperbarui ke rantai terpanjang atau paling valid yang mereka ketahui. Ketika blok baru diterima yang menghasilkan skor kumulatif yang lebih tinggi, sebuah node akan memperbarui salinan lokal blockchain-nya dan menyebarkan blok baru tersebut ke rekan-rekannya. Tidak ada kepastian mutlak bahwa transaksi tertentu akan tetap berada dalam versi blockchain yang diterima selamanya. Meskipun demikian, karena blok baru



umumnya ditambahkan ke blok yang sudah ada—dan penulisan ulang blok lama tidak menguntungkan secara komputasi—kemungkinan mengubah transaksi yang telah dikonfirmasi berkurang secara eksponensial dengan setiap blok tambahan. Sebagai contoh, dalam sistem blockchain yang berbasis pada mekanisme konsensus *Proof-of-Work*, rantai dengan pekerjaan kumulatif terbesar dianggap valid oleh jaringan. Beberapa teka-teki kriptografi dapat digunakan untuk membuktikan upaya komputasi yang cukup. Tidak seperti sistem terdistribusi tradisional, blockchain bergantung pada komputasi redundan, yang meningkatkan kekokohan dan kepercayaan.

Tanda Tangan Digital

Tanda tangan tulisan tangan telah lama memegang peran penting sebagai metode verifikasi identitas. Tanda tangan ini berfungsi sebagai sarana untuk mengautentikasi dokumen kertas dengan membuktikan bahwa individu tertentu, dan bukan orang lain, yang menandatangani. Untuk memverifikasi tanda tangan tulisan tangan, biasanya dibandingkan dengan tanda tangan individu yang telah dicatat sebelumnya. Jika tanda tangan cocok, penerima dapat dengan yakin menerima dokumen tersebut; jika tidak, dokumen tersebut dapat ditolak. Bagi penandatanganan pertama kali, identitas ditetapkan melalui identifikasi resmi dan kehadiran fisik selama penandatanganan. Ciri-ciri utama tanda tangan tulisan tangan meliputi:

- a. Tanda tangan harus secara unik mengidentifikasi individu tersebut.
- b. Setelah verifikasi, tanda tangan tersebut harus dapat dikaitkan dengan satu orang tertentu.

Tanda tangan digital berfungsi sebagai padanan elektronik dari tanda tangan tulisan tangan—mirip dengan sidik jari digital atau stempel elektronik yang ditempatkan pada dokumen. Meskipun keduanya memiliki tujuan yang sama—mengautentikasi asal dan integritas pesan—ada perbedaan penting: tanda tangan digital unik untuk setiap dokumen, bahkan jika penanda tangan dan pasangan kunci publik/pribadi mereka tetap sama. Selain itu, tanda tangan digital memastikan autentikasi asal data dan integritas pesan. Penggunaan fungsi hash dalam tanda tangan digital biasanya mengikuti proses ini:

- a. Misalkan Sarah adalah pengirim dan juga penandatanganan dokumen digital. Dia memiliki pasangan kunci publik/pribadi, dokumen yang akan dikirim, dan fungsi hash untuk menghitung ringkasan pesan.
- b. Untuk memulai penandatanganan, Sarah menghitung hash (ringkasan) dokumen tersebut.
- c. Kemudian, dia mengenkripsi ringkasan ini menggunakan kunci pribadinya, sehingga menciptakan tanda tangan digital.
- d. Tanda tangan digital ini ditambahkan ke dokumen asli.
- e. Secara opsional, Sarah dapat mengenkripsi dokumen yang telah ditandatangani untuk memastikan kerahasiaan sebelum mengirimkannya kepada penerima, Remy.

Setelah menerima ciphertext, Remy melakukan langkah-langkah berikut untuk memvalidasi dokumen:



- a. Dia mendekripsi ciphertext menggunakan **kunci publik** Sarah untuk memulihkan dokumen yang telah ditandatangani dan tanda tangannya.
 - b. Kemudian, dia menghitung hash dari dokumen yang diterima dan membandingkannya dengan tanda tangan yang telah didekripsi (yaitu, hash yang ditandatangani oleh Sarah).
 - c. Jika nilai-nilainya cocok, dokumen tersebut diautentikasi dan integritasnya diverifikasi.
- Proses ini memastikan bahwa pesan tersebut memang dikirim oleh Sarah dan tidak diubah selama pengiriman.

11.4 LATIHAN

1. **Strategi Kata Sandi Grafis:** Rancang sistem autentikasi yang aman yang memaksimalkan ruang kata sandi sekaligus meningkatkan daya ingat.
 - Implementasikan skema kata sandi grafis yang dikombinasikan dengan fungsi hash.
 - Manfaatkan ketahanan bawaan kata sandi grafis terhadap serangan tradisional seperti brute force, kamus, rekayasa sosial, dan spyware.
2. **Enkripsi File Menggunakan Deret Fibonacci:** Kembangkan mekanisme enkripsi-dekripsi berdasarkan deret Fibonacci.
 - Memulai deret Fibonacci dari 1, bukan 0.
 - Mengenkripsi file sedemikian rupa sehingga karakter pada posisi ganjil dalam ciphertext digeser ke depan menggunakan suku Fibonacci yang sesuai.
3. **Algoritma Enkripsi AES–DES Hibrida:** Membuat sistem enkripsi yang kuat dengan mengintegrasikan AES dan Standar Enkripsi Data (DES) dalam struktur terpadu.
 - Menggabungkan AES dan DES dalam arsitektur seperti Feistel yang efisien.
 - Meningkatkan kekuatan enkripsi dengan memanfaatkan keunggulan gabungan dari kedua algoritma.
4. **Proyek Enkripsi Mandiri Seluler:** Menerapkan sistem perlindungan data yang mengenkripsi sendiri untuk perangkat seluler menggunakan cipher aliran.
 - Menyimpan kunci enkripsi di server jarak jauh.
 - Jika ponsel hilang, pengguna dapat melaporkannya, sehingga server akan menghapus kunci yang terkait.
 - Ini memastikan data tetap rahasia meskipun perangkat diretas.

Catatan

1 Meskipun e biasanya dipilih secara acak, dalam praktiknya nilai-nilai kecil seperti $e = 3$ atau bilangan prima dalam bentuk $e = 2^i + 1$ (misalnya, 65537) sering digunakan untuk mengoptimalkan efisiensi eksponensiasi. Perhatikan bahwa karena $\varphi(N)$ genap, e harus ganjil dan lebih besar dari 1.



BAB 12

PENGANTAR APLIKASI DUNIA CERDAS

12.1 APLIKASI MENARIK

Aplikasi cerdas mentransformasi berbagai industri dengan meningkatkan efisiensi, memperbaiki pengambilan keputusan, dan mengatasi tantangan kompleks. Unit ini mengeksplorasi fungsionalitas, manfaat, dan hambatan yang ditimbulkannya. Unit ini mengkaji teknologi dasar seperti Internet of Things (IoT), kecerdasan buatan (AI), dan pembelajaran mesin (ML), menyoroti aplikasinya di bidang kesehatan, transportasi, pertanian, dan manajemen energi. Melalui contoh dan studi kasus dunia nyata, unit ini menunjukkan bagaimana teknologi ini diimplementasikan dan dampaknya terhadap masyarakat. Dengan menjembatani inovasi dengan kepraktisan, Unit 3 memberikan wawasan berharga tentang lanskap teknologi cerdas yang terus berkembang dan perannya dalam membentuk masa depan. Tujuan dari bab ini adalah sebagai berikut:

- **Mengkatalogkan Aplikasi Dunia Cerdas yang Signifikan:** Untuk mencantumkan berbagai aplikasi dunia cerdas yang penting dalam kerangka kerja terpadu aplikasi algoritmik.
- **Memperdalam Pemahaman tentang Teknologi dan Tantangan:** Untuk memberikan wawasan yang lebih dalam tentang aplikasi cerdas dengan memeriksa teknologi yang digunakan dan mengidentifikasi hambatan yang terkait dengan kompleksitas.
- **Perkembangan Masa Depan dan Ide Proyek:** Untuk memperkaya pembaca dengan potensi perkembangan masa depan dan menginspirasi ide proyek di bidang aplikasi cerdas.
- **Paradigma Algoritmik Standar:** Ini adalah model atau kerangka kerja generik yang mendasari desain suatu kelas algoritma.
- **Masalah Cerdas:** Ini adalah masalah yang didefinisikan secara ringkas yang menguraikan masalah tanpa membahas penyebab atau solusinya.
- **Solusi Cerdas:** Istilah ini mengacu pada sistem yang mengintegrasikan teknologi inovatif dalam perangkat keras (misalnya, IoT) dan perangkat lunak.
- **Algoritma Tepat dan Perkiraan:** Bagian ini membahas bagaimana algoritma tepat menemukan solusi yang tepat, sementara algoritma perkiraan memberikan solusi ketika jawaban tepat tidak dapat diperoleh. Terakhir, bagian ini mengeksplorasi bagaimana hasil berbasis data diperoleh.

Aplikasi dunia cerdas mencakup berbagai teknologi inovatif, termasuk mesin pencari web canggih (misalnya, Google), sistem rekomendasi (digunakan oleh platform seperti YouTube, Amazon, dan Netflix), sistem pengenalan suara (seperti Siri dan Alexa), pengambilan keputusan otomatis mobil otonom, dan kemampuan bermain game strategis yang terlihat pada sistem seperti catur dan Go. Seiring kemajuan perangkat keras, tugas-tugas yang



membutuhkan "kecerdasan" menjadi semakin umum. Bagian iii buku ini mengilustrasikan berbagai aplikasi algoritma dunia cerdas, seperti yang dibahas di bagian i dan ii.

Bagian iii merinci penerapan algoritma yang tangguh ini dalam kerangka analisis algoritma standar. Algoritma dunia cerdas yang paling signifikan dan praktis ditunjukkan, dianalisis, dan diterapkan. Aplikasi yang dibahas dalam bagian ini meliputi pendidikan cerdas, pertanian cerdas, transportasi cerdas, pemerintahan cerdas, dan manajemen bencana untuk kota cerdas. Bab ini menguraikan peta jalan untuk bab-bab selanjutnya dalam buku ini, dengan fokus pada algoritma kunci dan aplikasinya dalam berbagai teknologi cerdas.

Bab 13 memperkenalkan kemajuan utama dalam teknologi pendidikan, menekankan peran ai dan analitik data dalam mentransformasi lingkungan pembelajaran. Bab ini mencakup area-area penting seperti pembelajaran personalisasi, penyampaian konten cerdas, analitik dan wawasan pembelajaran, analitik data dalam pendidikan, tutor dan asisten ai dalam pendidikan, serta penilaian dan umpan balik dalam pendidikan. Dengan memanfaatkan pendekatan berbasis ai, lembaga pendidikan dapat menciptakan pengalaman pembelajaran personalisasi yang disesuaikan dengan kebutuhan individu siswa, mengoptimalkan penyampaian konten melalui platform pembelajaran adaptif, dan mengekstrak wawasan yang bermakna dari kumpulan data pendidikan yang luas. Tutor ai dan asisten virtual meningkatkan pembelajaran interaktif, memberikan dukungan waktu nyata kepada siswa, sementara mekanisme penilaian dan umpan balik tingkat lanjut meningkatkan hasil pembelajaran. Teknologi-teknologi ini secara kolektif berkontribusi pada kerangka kerja pendidikan yang lebih cerdas, responsif, dan efektif.

Bab 14 mengeksplorasi peran algoritma dalam perawatan kesehatan cerdas, membahas aspek-aspek penting dari penjadwalan alur pasien, pengemasan obat, keamanan data, dan pemantauan nutrisi otomatis. Bab ini dimulai dengan penjadwalan alur pasien dan perencanaan kapasitas, memperkenalkan teori antrian, algoritma simulasi, dan teknik pemrograman linier untuk mengoptimalkan efisiensi layanan perawatan kesehatan. Kemudian, bab ini beralih ke sektor pengemasan obat dalam industri perawatan kesehatan, mencakup otomatisasi proses robotik, pengenalan karakter optik, dan analitik prediktif untuk menyederhanakan pengemasan dan distribusi obat. Bab ini juga menyoroti pentingnya keamanan data dalam perawatan kesehatan cerdas dengan membahas algoritma enkripsi, teknologi blockchain, dan teknik ml untuk deteksi anomali. Terakhir, sistem pemantauan nutrisi otomatis dieksplorasi, yang menggabungkan algoritma penilaian diet, sistem rekomendasi, dan teknologi pengenalan gambar untuk meningkatkan perawatan kesehatan pribadi dan pelacakan kesehatan.

Bab 15 mengeksplorasi pendekatan algoritmik modern dalam aplikasi lingkungan, membahas area kritis seperti pemantauan polusi udara perkotaan, penjadwalan energi hijau, jaringan cerdas, pengelolaan limbah cerdas, dan pemantauan drone. Bab ini dimulai dengan crowdsensing untuk pemantauan polusi udara perkotaan, merinci algoritma kunci dan memberikan contoh aplikasi. Kemudian, bab ini membahas penjadwalan energi hijau untuk manajemen sisi permintaan, dengan fokus pada teknik optimasi untuk menyeimbangkan konsumsi energi secara efisien. Konsep jaringan cerdas juga dibahas, menekankan algoritma



kunci dan aplikasi dunia nyata yang meningkatkan distribusi dan konsumsi energi. Selanjutnya, bab ini membahas sistem pengelolaan limbah cerdas, menyoroti algoritma untuk strategi pengumpulan, pemantauan, dan pembuangan limbah yang efisien. Bagian terakhir bab ini membahas pemantauan menggunakan drone, menampilkan algoritma kunci dan contoh aplikasi yang menggambarkan peran drone dalam pengawasan lingkungan dan pengumpulan data. Melalui pembahasan ini, pembaca akan memperoleh wawasan tentang bagaimana teknik komputasi modern berkontribusi pada keberlanjutan dan efisiensi lingkungan.

Bab 16 mengeksplorasi peran pertanian cerdas dan algoritma terkaitnya, mencakup aspek-aspek penting seperti pertanian presisi, pemantauan kesehatan tanah, manajemen irigasi, dan prediksi hasil panen. Bab ini dimulai dengan pertanian presisi, menyoroti algoritma kunci dan contoh aplikasinya dalam mengoptimalkan pemanfaatan sumber daya dan manajemen tanaman. Selanjutnya, pemantauan kesehatan tanah diperiksa, merinci algoritma yang menilai kualitas tanah dan memprediksi kekurangan nutrisi, bersama dengan contoh implementasi praktis. Manajemen irigasi kemudian dibahas, menekankan algoritma kunci yang meningkatkan efisiensi distribusi air dan mengurangi pemborosan melalui sistem otomatis. Bab ini juga membahas prediksi hasil panen, menampilkan model ml dan statistik yang digunakan untuk memperkirakan produktivitas tanaman berdasarkan data lingkungan dan historis. Melalui diskusi ini, pembaca akan memperoleh wawasan tentang bagaimana teknik komputasi canggih berkontribusi pada praktik pertanian yang berkelanjutan dan efisien.

Bab 17 mengeksplorasi manajemen lalu lintas cerdas dan sistem transportasi cerdas (its), mencakup aplikasi yang meningkatkan mobilitas perkotaan, transportasi umum, dan efisiensi infrastruktur. Bab ini dimulai dengan manajemen lalu lintas cerdas, menyajikan algoritma dan aplikasi kunci yang mengoptimalkan aliran lalu lintas dan mengurangi kemacetan. Kemudian membahas its, membahas pendekatan berbasis data untuk meningkatkan jaringan transportasi. Peran algoritma dalam transportasi publik juga dibahas, dengan fokus pada optimasi rute dan teknik penjadwalan waktu nyata. Sistem parkir pintar dieksplorasi selanjutnya, menyoroti solusi inovatif untuk mengelola parkir perkotaan dan mengurangi emisi kendaraan. Bab ini juga membahas kendaraan otonom, membahas algoritma navigasi berbasis ml dan sensor. Pemantauan dan pemeliharaan infrastruktur dibahas, menampilkan analitik prediktif untuk pemeliharaan jalan dan jembatan. Kendaraan listrik dan terhubung kemudian dibahas, menekankan perutean hemat energi dan komunikasi antar kendaraan. Terakhir, bab ini diakhiri dengan strategi tanggap darurat, menyajikan algoritma yang meningkatkan koordinasi dan mengurangi waktu respons dalam situasi kritis. Melalui diskusi ini, pembaca akan memperoleh pemahaman komprehensif tentang bagaimana algoritma canggih berkontribusi pada masa depan mobilitas perkotaan dan transportasi pintar.

Bab 18 mengeksplorasi persimpangan ti dan masyarakat, menyoroti bagaimana teknologi berkontribusi pada tantangan dan kemajuan masyarakat. Bab ini dimulai dengan pengantar peran ti dalam sistem sosial. Kemudian membahas identifikasi orang hilang, membahas algoritma kunci dan aplikasi dunia nyata yang memanfaatkan pengenalan wajah dan pelacakan berbasis ai untuk menemukan individu. Bagian selanjutnya membahas penalaran sosial, mengeksplorasi bagaimana informasi, perilaku, dan tren menyebar melalui



jaringan sosial, dengan fokus pada teknik pemodelan prediktif. Penyebaran penyakit kemudian dibahas, merinci model epidemiologi dan metode komputasi yang digunakan untuk memahami dan mengurangi penyebaran penyakit. Bab ini juga mengeksplorasi pemantauan kejahatan, menampilkan algoritma yang dirancang untuk menganalisis pola kejahatan, mendeteksi anomali, dan meningkatkan strategi penegakan hukum. Terakhir, bab ini menyajikan proyek-proyek yang sedang berlangsung yang menerapkan solusi ti untuk tantangan masyarakat, menunjukkan dampak transformatif teknologi di berbagai bidang. Melalui diskusi ini, pembaca akan memperoleh wawasan tentang bagaimana ti memengaruhi masyarakat dan berkontribusi dalam memecahkan masalah sosial yang kompleks.

Bab 19 mengeksplorasi peran pemerintahan cerdas, berfokus pada transformasi digital dalam tata kelola melalui teknologi canggih. Bab ini dimulai dengan layanan e-government, membahas algoritma kunci yang digunakan untuk autentikasi yang aman, manajemen data warga, dan penyampaian layanan publik otomatis. Kemudian membahas utilitas cerdas, menampilkan algoritma yang mengoptimalkan distribusi energi, manajemen air, dan pengendalian limbah untuk operasi perkotaan yang efisien. Bab ini juga membahas peningkatan keselamatan publik, dengan merinci algoritma kepolisian prediktif dan respons darurat yang meningkatkan langkah-langkah keselamatan. Kemudian, bab ini membahas pemantauan lingkungan, menyoroti algoritma untuk pelacakan kualitas udara, penilaian dampak iklim, dan prediksi risiko bencana. Terakhir, proyek-proyek yang sedang berlangsung di bidang pemerintahan cerdas dipresentasikan, yang menunjukkan implementasi nyata dari solusi tata kelola berbasis ai yang meningkatkan transparansi, efisiensi, dan keterlibatan publik.

Bab 20 mengeksplorasi beragam aplikasi dan algoritma canggih yang digunakan dalam manajemen bencana, meliputi komunikasi pasca-kecelakaan tambang, manajemen informasi bencana, dan jaringan sensor cerdas untuk mitigasi bencana. Bab ini membahas sistem leaky-feeder untuk meningkatkan komunikasi pasca-kecelakaan tambang dan penambangan data untuk manajemen informasi bencana, merinci arsitektur sistemnya, dasbor dinamis, dan teknik pembangkitan komunitas. Selain itu, bab ini membahas algoritma untuk jaringan sensor cerdas, menjelaskan bagaimana jaringan sensor nirkabel (wsn) berfungsi dalam manajemen bencana. Terakhir, bab ini menyoroti proyek-proyek penting, termasuk dasbor pemantauan bencana waktu nyata, deteksi epidemi menggunakan ml, manajemen risiko gempa bumi, dan sistem pemulihan bencana berbasis komunitas.

Bab 21 mengeksplorasi algoritma komunikasi, meliputi strategi komunikasi penting di berbagai lingkungan jaringan. Bab ini dimulai dengan algoritma komunikasi untuk wsn, membahas algoritma dan aplikasi utama untuk transmisi data yang efisien. Selanjutnya, bab ini mengeksplorasi algoritma komunikasi berbasis store-carry-forward untuk jaringan yang toleran terhadap penundaan, menyoroti teknik-teknik utama untuk konektivitas intermiten. Bab ini juga membahas algoritma komunikasi berbasis daya rendah untuk jaringan berdaya rendah dan rentan terhadap kehilangan data, dengan fokus pada metode komunikasi hemat energi. Algoritma jaringan yang didefinisikan perangkat lunak (software-defined networking/sdn) dibahas selanjutnya, dengan menekankan perannya dalam manajemen



jaringan yang fleksibel dan terprogram. Algoritma jaringan peer-to-peer kemudian diperiksa, dengan membahas strategi komunikasi terdesentralisasi. Bab ini diakhiri dengan proyek-proyek yang sedang berlangsung dalam algoritma komunikasi, yang menampilkan aplikasi dunia nyata dalam mengoptimalkan kinerja jaringan.



BAB 13

PENDIDIKAN CERDAS

13.1 CONTOH ALAT PENDIDIKAN CERDAS

Perkembangan pesat pendidikan daring dan pembelajaran elektronik telah menghasilkan aliran data pendidikan yang sangat besar. Masuknya data secara terus-menerus ini menghadirkan tantangan signifikan dalam mengidentifikasi sumber belajar yang tepat dan mitra belajar yang sesuai. Bab ini membahas tantangan-tantangan ini dengan menyediakan aplikasi pendidikan cerdas yang berkelanjutan. Aplikasi-aplikasi ini mencakup pembelajaran personal, rekomendasi, dan pengelompokan adaptif dalam berbagai konteks pendidikan. Tujuan bab ini adalah sebagai berikut.

- **Pembelajaran yang Dipersonalisasi:** Mengembangkan pengalaman pendidikan yang disesuaikan dengan kebutuhan dan preferensi masing-masing peserta didik.
- **Analisis Data:** Memanfaatkan analisis data tingkat lanjut untuk mendapatkan wawasan dari data besar pendidikan, memfasilitasi pengambilan keputusan dan pengembangan strategi yang tepat.
- **Tutor dan Asisten AI:** Menerapkan tutor dan asisten berbasis AI untuk memberikan dukungan dan bimbingan secara waktu nyata, meningkatkan pengalaman belajar.
- **Pembuatan Konten:** Memanfaatkan algoritma untuk membuat dan mengelola konten pendidikan yang relevan, menarik, dan selaras dengan tujuan pembelajaran.
- **Penilaian dan Umpan Balik:** Menggunakan alat penilaian cerdas untuk mengevaluasi kinerja peserta didik dan memberikan umpan balik yang membangun, mendorong peningkatan berkelanjutan.
- **Pembelajaran Kolaboratif:** Memfasilitasi lingkungan pembelajaran kolaboratif di mana peserta didik dapat berinteraksi, berbagi pengetahuan, dan bekerja sama menuju tujuan bersama.

Beberapa alat pendidikan cerdas memanfaatkan algoritma canggih untuk meningkatkan pengalaman belajar dan menyediakan pendidikan yang dipersonalisasi. Khan Academy menggunakan analitik data dan algoritma pembelajaran adaptif untuk menciptakan pengalaman belajar yang dipersonalisasi dengan umpan balik instan, memastikan siswa maju sesuai dengan kecepatan mereka sendiri. Duolingo menggunakan teknik pembelajaran adaptif dan gamifikasi untuk menyesuaikan platform pembelajaran bahasanya sesuai dengan kinerja pengguna, membuat pembelajaran lebih menarik dan efektif. Coursera mengintegrasikan sistem rekomendasi dan analitik prediktif untuk menawarkan rekomendasi kursus yang dipersonalisasi dan jalur pembelajaran yang disesuaikan, membantu pelajar menemukan konten yang paling relevan. Smart Sparrow menyediakan alat pembelajaran adaptif dan pembuatan konten, memungkinkan pendidik untuk merancang pengalaman belajar yang dipersonalisasi berdasarkan interaksi siswa. Alat-alat ini menunjukkan bagaimana platform



pendidikan berbasis teknologi merevolusi lanskap pembelajaran. Detail alat pendidikan cerdas ditunjukkan pada Tabel 13.1.

TABEL 13.1 Contoh alat pendidikan cerdas dan algoritmanya.

Alat pendidikan cerdas	Algoritma yang digunakan
Khan Academy	Analisis data, pembelajaran adaptif
Duolingo	Pembelajaran adaptif, gamifikasi
Coursera	Sistem rekomendasi, analitik prediktif
Smart Sparrow	Pembelajaran adaptif, alat pembuat konten
EdX	Pembelajaran mesin, jalur pembelajaran yang dipersonalisasi
Socratic	Penjawab pertanyaan bertenaga AI, pemrosesan bahasa alami
Quizlet	Rekomendasi kartu pengingat (<i>flashcard</i>) berbasis AI, pengujian adaptif
Google Classroom	Kolaborasi berbasis awan (<i>cloud</i>), penilaian otomatis
Blackboard Learn	Analitik pembelajaran, wawasan bertenaga AI

13.2 PEMBELAJARAN YANG DIPERSONALISASI

Pendidikan cerdas memanfaatkan algoritma canggih dari konsep yang lebih luas tentang “dunia cerdas” untuk menciptakan pengalaman belajar yang dipersonalisasi dan adaptif. Pembelajaran yang dipersonalisasi dalam pendidikan cerdas menyesuaikan pengalaman pendidikan dengan kebutuhan, preferensi, dan kecepatan belajar individu. Pendekatan ini meningkatkan keterlibatan siswa, meningkatkan hasil belajar, dan mengakomodasi beragam gaya belajar dengan memanfaatkan algoritma canggih. Bagian ini memberikan tinjauan mendalam tentang mekanisme dan algoritma di balik pembelajaran yang dipersonalisasi. Sistem pembelajaran adaptif secara dinamis menyesuaikan konten dan tingkat kesulitan berdasarkan kinerja dan perilaku peserta didik. Sistem ini memastikan bahwa setiap siswa menerima tingkat tantangan dan dukungan yang tepat.

Algoritma Utama

- **Pemodelan Pengguna:** Membuat profil peserta didik berdasarkan interaksi, kinerja, dan preferensi mereka melibatkan pengumpulan dan analisis data tentang bagaimana peserta didik berinteraksi dengan materi pendidikan, tingkat keberhasilan mereka, dan minat mereka. Seperti yang dibahas oleh Brusilovsky, proses ini menggunakan berbagai teknik analitik data dan AI untuk melacak aktivitas peserta didik, seperti waktu yang dihabiskan untuk tugas, partisipasi dalam diskusi, dan frekuensi penggunaan sumber daya. Metrik kinerja, termasuk nilai ujian dan nilai tugas, diintegrasikan untuk mengukur kemahiran dan area yang perlu ditingkatkan. Selain itu, preferensi disimpulkan dari pilihan yang dibuat dalam jalur pembelajaran, pemilihan konten, dan umpan balik. Informasi ini dibangun dengan mensintesis profil pembelajar yang komprehensif dan dinamis, memungkinkan pengalaman pendidikan yang dipersonalisasi yang disesuaikan dengan gaya dan kebutuhan belajar unik individu.



- **Pembelajaran Mesin (ML):** Menganalisis data untuk memprediksi kebutuhan pembelajar dan menyesuaikan konten sesuai dengan itu. Algoritma ML memproses sejumlah besar data pendidikan, mengidentifikasi pola dan tren dalam perilaku, kinerja, dan preferensi pembelajar. Dengan terus belajar dari data ini, algoritma dapat meramalkan kebutuhan pembelajar di masa depan, seperti mengidentifikasi topik mana yang mungkin mereka kesulitan atau sumber daya mana yang kemungkinan besar akan mereka anggap paling bermanfaat. Kemampuan prediktif ini memungkinkan penyesuaian konten yang dinamis, memastikan bahwa setiap pembelajar menerima pengalaman pendidikan yang dipersonalisasi yang berkembang sebagai respons terhadap interaksi dan pencapaian mereka yang berkelanjutan.
- **Jaringan Bayesian:** Model probabilistik yang membantu membuat keputusan dalam ketidakpastian, menyesuaikan konten berdasarkan kemungkinan berbagai hasil pembelajaran. Jaringan ini menggunakan struktur grafis untuk merepresentasikan dan menghitung hubungan probabilistik di antara serangkaian variabel, seperti karakteristik pembelajar, perilaku, dan metrik kinerja. Dengan memperbarui probabilitas saat data baru tersedia, jaringan Bayesian dapat menyimpulkan hasil pembelajaran yang paling mungkin dan menyarankan penyesuaian konten yang dipersonalisasi untuk mengoptimalkan pengalaman pendidikan pembelajar. Pendekatan ini memungkinkan pengambilan keputusan yang lebih informatif dan adaptif dalam lingkungan pendidikan, memastikan bahwa penyampaian konten responsif dan relevan dengan kebutuhan individu pembelajar.

Contoh Aplikasi

Brusilovsky dan Millán menjelaskan sistem hipermedia adaptif yang mempersonalisasi konten pendidikan dengan menyesuaikan tingkat kesulitan dan gaya presentasi berdasarkan kinerja pembelajar yang sedang berlangsung. Sistem ini memanfaatkan model terperinci tentang pengetahuan, preferensi, dan perilaku pembelajar untuk menyesuaikan pengalaman belajar secara dinamis. Saat pembelajar berinteraksi dengan sistem, sistem terus memantau kemajuan mereka dan menyesuaikan konten secara waktu nyata agar sesuai dengan kebutuhan mereka yang terus berkembang. Misalnya, jika seorang pembelajar secara konsisten unggul dalam bidang tertentu, sistem dapat meningkatkan kesulitan materi selanjutnya untuk memberikan tantangan yang lebih besar. Sebaliknya, jika pembelajar kesulitan, sistem dapat menyederhanakan konten atau mengubah gaya penyajian untuk meningkatkan pemahaman. Pendekatan personalisasi ini bertujuan untuk mengoptimalkan efisiensi dan keterlibatan pembelajaran dengan menyediakan jalur pendidikan yang disesuaikan yang merespons kemampuan dan kebutuhan unik setiap pembelajar.

13.3 PENYAMPAIAN KONTEN CERDAS

Penyampaian konten cerdas melibatkan penggunaan algoritma untuk menyajikan materi pendidikan dengan cara yang paling sesuai dengan kondisi dan gaya belajar pembelajar saat ini. Pendekatan ini menyesuaikan pengalaman belajar dengan mengadaptasi konten berdasarkan kemajuan dan preferensi individu, memastikan bahwa setiap siswa menerima



informasi dengan cara yang paling efektif untuk kebutuhan unik mereka. Dengan terus menganalisis data tentang interaksi dan kinerja siswa, algoritma ini dapat secara dinamis menyesuaikan jenis dan kesulitan materi yang disajikan. Metode personalisasi ini tidak hanya meningkatkan keterlibatan dan pemahaman tetapi juga mengoptimalkan keseluruhan proses pembelajaran, membuat pendidikan lebih efisien dan efektif.

Algoritma Utama

- **Pemrosesan Bahasa Alami (NLP):** Menganalisis teks dan ucapan untuk memahami dan menghasilkan respons dan penjelasan seperti manusia.
- **Pembelajaran Penguatan (RL):** Mengoptimalkan penyampaian konten berdasarkan umpan balik berkelanjutan dari interaksi pembelajar.

Contoh Aplikasi

Sistem pembuatan konten otomatis menggunakan NLP untuk membuat penjelasan dan contoh khusus yang sesuai dengan tingkat pemahaman pembelajar saat ini.

13.4 ANALISIS DAN WAWASAN PEMBELAJARAN

Analisis pembelajaran melibatkan pengumpulan dan analisis data tentang interaksi, kinerja, dan keterlibatan siswa untuk mendapatkan wawasan tentang proses dan hasil pembelajaran mereka. Dengan memanfaatkan data ini, pendidik dapat mengidentifikasi pola dan tren yang mengungkapkan bagaimana siswa belajar, di mana mereka mengalami kesulitan, dan strategi apa yang paling efektif. Pendekatan berbasis data ini memungkinkan pengembangan pengalaman belajar yang dipersonalisasi, karena wawasan yang diperoleh dapat digunakan untuk menyesuaikan intervensi pendidikan untuk memenuhi kebutuhan spesifik setiap siswa. Selain itu, analitik pembelajaran dapat membantu lembaga meningkatkan kurikulum dan metode pengajaran mereka, yang pada akhirnya mengarah pada hasil pendidikan yang lebih baik.

Algoritma Utama

- **Penambangan Data:** Mengekstrak pola dan tren yang berguna dari kumpulan data besar.
- **Pengelompokan dan Klasifikasi:** Mengelompokkan peserta didik dengan perilaku dan karakteristik yang serupa, memungkinkan intervensi yang ditargetkan.

Contoh Aplikasi

Siemens dan Long menyoroti penggunaan analitik pembelajaran untuk mengidentifikasi siswa yang berisiko dan memberikan intervensi tepat waktu.

13.5 ANALITIK DATA DALAM PENDIDIKAN

Analitik data dalam pendidikan melibatkan pemanfaatan teknik analitik data tingkat lanjut untuk mendapatkan wawasan dari big data pendidikan. Proses ini memfasilitasi pengambilan keputusan dan pengembangan strategi yang terinformasi, meningkatkan hasil pendidikan dan pengalaman belajar yang dipersonalisasi. Integrasi algoritma dunia cerdas memungkinkan analisis sejumlah besar data yang dihasilkan di lingkungan pendidikan, memberikan wawasan yang dapat ditindaklanjuti bagi pendidik dan administrator.



Algoritma Utama

Beberapa algoritma utama digunakan dalam analitik data pendidikan, memanfaatkan kemampuan algoritma dunia cerdas untuk memproses dan menganalisis big data pendidikan:

- **Algoritma ML:** Algoritma ini, termasuk pohon keputusan, mesin vektor pendukung, dan jaringan saraf, digunakan untuk mengidentifikasi pola dan memprediksi hasil berdasarkan data historis.
- **NLP:** Algoritma NLP menganalisis data teks dari interaksi siswa, tugas, dan umpan balik untuk mengekstrak wawasan dan tren yang bermakna.
- **RL:** Pendekatan ini melibatkan pengoptimalan jalur pembelajaran dan penyampaian konten berdasarkan umpan balik berkelanjutan dari interaksi siswa.

Contoh Aplikasi

Contoh aplikasi analitik data tingkat lanjut dalam pendidikan menggunakan algoritma dunia cerdas adalah pengembangan platform pembelajaran adaptif. Platform ini memanfaatkan data dari interaksi siswa, metrik kinerja, dan umpan balik untuk terus beradaptasi dan mempersonalisasi pengalaman belajar. Misalnya, sistem pembelajaran adaptif dapat menggunakan algoritma ML untuk memprediksi topik mana yang sulit dipahami siswa dan menyesuaikan penyampaian konten sesuai dengan itu. Selain itu, NLP dapat digunakan untuk menganalisis umpan balik siswa dan meningkatkan materi pembelajaran. Algoritma RL lebih lanjut meningkatkan platform dengan mengoptimalkan urutan konten pendidikan berdasarkan keterlibatan siswa dan tingkat keberhasilan.

13.6 TUTOR DAN ASISTEN AI DALAM PENDIDIKAN

Tutor dan asisten AI mentransformasi pendidikan dengan menyediakan pengalaman belajar yang dipersonalisasi, mengotomatiskan tugas administratif, dan menawarkan dukungan waktu nyata kepada siswa dan pendidik. Sistem AI ini memanfaatkan algoritma dunia cerdas untuk beradaptasi dengan kebutuhan masing-masing pembelajar, memastikan bahwa konten pendidikan disesuaikan dengan kecepatan dan gaya belajar mereka.

Algoritma Utama

Tutor dan asisten AI menggunakan berbagai algoritma untuk memberikan dukungan pendidikan yang efektif. Algoritma NLP memungkinkan tutor AI untuk memahami dan menanggapi pertanyaan siswa dalam bahasa alami, memfasilitasi pengalaman belajar interaktif. Algoritma ML menganalisis data siswa untuk memprediksi kebutuhan belajar dan menyesuaikan strategi pengajaran sesuai dengan kebutuhan tersebut. Algoritma RL mengoptimalkan proses bimbingan dengan terus meningkatkan kinerja AI berdasarkan umpan balik dan hasil belajar siswa.

Contoh Aplikasi

Contoh tutor dan asisten AI dalam pendidikan adalah penggunaan sistem bimbingan cerdas (ITS). Sistem ini menggunakan ML dan NLP untuk menciptakan lingkungan belajar yang dinamis dan interaktif. Misalnya, ITS dapat memberikan umpan balik yang dipersonalisasi pada tugas siswa, mengidentifikasi area di mana siswa kesulitan, dan menawarkan latihan yang ditargetkan untuk meningkatkan pemahaman. Selain itu, algoritma RL dapat menyesuaikan



tingkat kesulitan pertanyaan berdasarkan kinerja siswa, memastikan kurva pembelajaran yang optimal.

13.7 EVALUASI PENDIDIKAN TERPADU

Penilaian dan umpan balik merupakan komponen penting dari proses pendidikan, memberikan wawasan tentang pembelajaran siswa dan memandu strategi pengajaran. Dengan memanfaatkan algoritma dunia cerdas, proses ini dapat ditingkatkan secara signifikan, menawarkan umpan balik yang waktu nyata, personal, dan dapat ditindaklanjuti kepada siswa dan pendidik. Pendekatan ini memanfaatkan analitik big data dan ML untuk terus memantau dan menilai kinerja siswa, memastikan intervensi dan dukungan tepat waktu.

Algoritma Utama

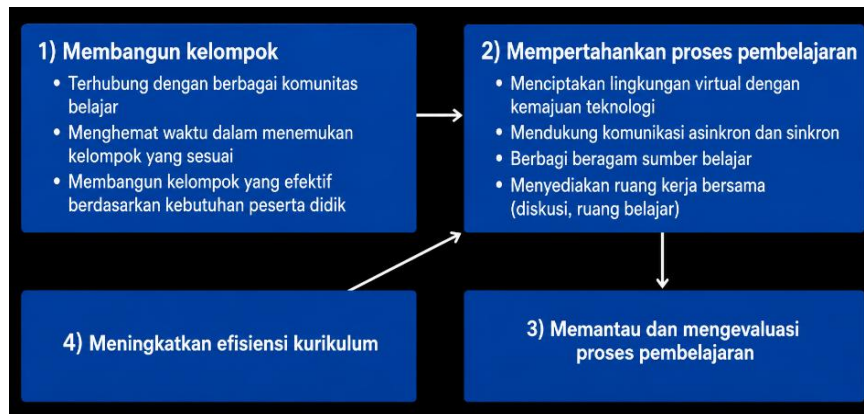
Algoritma dunia cerdas menggunakan beberapa teknik utama untuk meningkatkan penilaian dan umpan balik dalam pendidikan. Algoritma NLP digunakan untuk menganalisis respons siswa dan menghasilkan umpan balik yang bermakna. Algoritma ML memprediksi kinerja siswa dan mengidentifikasi area yang perlu ditingkatkan dengan menganalisis data historis dan interaksi saat ini. Algoritma RL mengoptimalkan proses umpan balik dengan beradaptasi dengan kebutuhan individu dan gaya belajar siswa berdasarkan data interaksi berkelanjutan.

Contoh Aplikasi

Contoh aplikasi algoritma dunia cerdas dalam penilaian dan umpan balik adalah implementasi sistem pengujian adaptif. Sistem ini memanfaatkan ML dan RL untuk menyesuaikan penilaian sesuai dengan pemahaman dan tingkat kinerja siswa saat ini. Misalnya, platform pengujian adaptif dapat menyesuaikan kesulitan pertanyaan secara waktu nyata, memberikan soal yang lebih mudah atau lebih menantang berdasarkan jawaban siswa. Algoritma NLP dapat menganalisis respons terbuka dan memberikan umpan balik instan dan terperinci, membantu siswa memahami kesalahan mereka dan meningkatkan hasil belajar mereka.

13.8 PEMBELAJARAN KOLABORATIF

Pembelajaran kolaboratif melibatkan siswa yang bekerja sama untuk memecahkan masalah, menyelesaikan tugas, atau memahami konsep baru. Dengan munculnya algoritma dunia cerdas, pembelajaran kolaboratif telah ditingkatkan dengan memanfaatkan analitik data, ML, dan NLP untuk menciptakan pengalaman belajar yang lebih efektif dan personal. Tahapan pembelajaran kolaboratif ditunjukkan pada Gambar 13.1. Teknologi ini memfasilitasi kolaborasi waktu nyata, memberikan wawasan tentang dinamika kelompok, dan membantu pendidik menyesuaikan strategi pengajaran mereka untuk meningkatkan hasil belajar siswa.



Gambar 13.1 Pembelajaran kolaboratif dan E-mentoring dalam lingkungan pembelajaran cerdas.

Algoritma Utama

Algoritma dunia cerdas menggunakan beberapa teknik utama untuk mendukung dan meningkatkan pembelajaran kolaboratif. Algoritma ML menganalisis pola interaksi antar siswa untuk mengidentifikasi strategi kolaborasi yang efektif dan area yang perlu ditingkatkan. Algoritma NLP digunakan untuk memproses dan memahami komunikasi antar siswa, memberikan umpan balik secara waktu nyata dan memfasilitasi komunikasi yang lebih baik. Selain itu, algoritma analisis jaringan sosial memeriksa struktur dan dinamika interaksi siswa untuk mengidentifikasi siswa yang berpengaruh dan mengoptimalkan pembentukan kelompok.

Contoh Aplikasi

Contoh aplikasi algoritma dunia cerdas dalam pembelajaran kolaboratif adalah penggunaan platform kolaboratif cerdas. Platform ini menggunakan ML untuk memantau dan menganalisis interaksi siswa, memberikan wawasan tentang dinamika kelompok dan kontribusi individu. Misalnya, platform kolaboratif cerdas dapat menggunakan NLP untuk memfasilitasi diskusi dengan menyoroti poin-poin penting, meringkas percakapan, dan memberikan umpan balik secara waktu nyata untuk memastikan interaksi yang produktif. Algoritma analisis jaringan sosial dapat digunakan untuk membentuk kelompok optimal berdasarkan kekuatan, kelemahan, dan pola interaksi siswa, sehingga meningkatkan pengalaman belajar secara keseluruhan.

13.9 LATIHAN

1. Apa itu pembelajaran personalisasi, dan bagaimana hal itu meningkatkan keterlibatan siswa dan hasil belajar dalam pendidikan cerdas?
2. Bagaimana algoritma ML memprediksi kebutuhan peserta didik dan menyesuaikan konten sesuai kebutuhan dalam sistem pembelajaran personalisasi?
3. Apa itu penyaringan kolaboratif, dan bagaimana cara kerjanya untuk memberikan rekomendasi personalisasi dalam konteks pendidikan?
4. Bagaimana metode penyaringan berbasis konten berbeda dari penyaringan kolaboratif, dan apa keunggulan masing-masing dalam sistem rekomendasi?



5. Bagaimana jaringan Bayesian dapat digunakan dalam sistem pembelajaran adaptif untuk membuat keputusan dalam kondisi ketidakpastian?
6. Kembangkan model pengguna untuk platform pembelajaran daring hipotetis, yang menggabungkan data tentang interaksi, kinerja, dan preferensi peserta didik.
7. Implementasikan algoritma penyaringan kolaboratif untuk merekomendasikan materi pembelajaran kepada siswa berdasarkan data interaksi mereka.
8. Buat model RL untuk mengoptimalkan proses bimbingan belajar berdasarkan umpan balik siswa.
9. Menerapkan sistem pengujian adaptif yang menyesuaikan penilaian dengan tingkat pemahaman dan kemajuan masing-masing siswa.
10. Mengembangkan algoritma NLP untuk menganalisis respons siswa dan menghasilkan umpan balik yang bermakna dalam sistem bimbingan belajar berbasis AI.



BAB 14

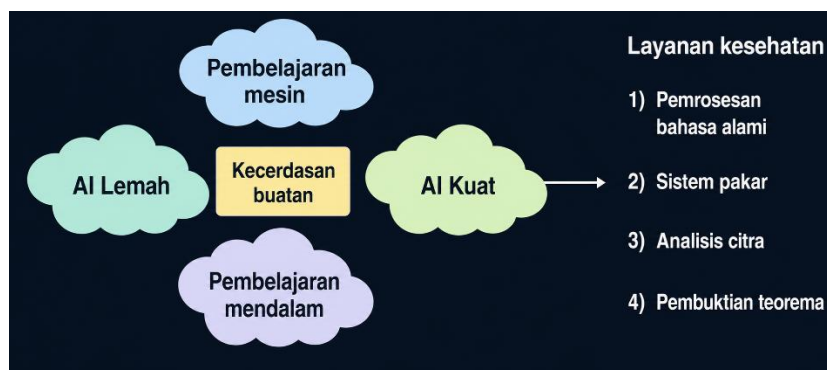
ALGORITMA MEDIS CERDAS

14.1 PENJADWALAN KAPASITAS PELAYANAN

Integrasi algoritma modern dalam perawatan kesehatan telah merevolusi bidang ini, memungkinkan perawatan pasien yang lebih cerdas, efisien, dan personal. Penerapan kecerdasan buatan (AI), pembelajaran mesin (ML), dan analitik data telah memberikan para profesional perawatan kesehatan alat yang ampuh untuk mendiagnosis penyakit, memprediksi hasil, dan menyesuaikan perawatan untuk setiap pasien. Bab ini mengeksplorasi algoritma kunci yang digunakan dalam perawatan kesehatan cerdas dan berbagai aplikasinya. Aplikasi yang dibahas dalam bab ini adalah:

- Penjadwalan alur pasien dan perencanaan kapasitas
- Pengemasan obat dalam industri perawatan kesehatan
- Keamanan data perawatan kesehatan cerdas

Penjadwalan alur pasien dan perencanaan kapasitas merupakan aspek penting dalam manajemen layanan kesehatan. Proses ini memastikan bahwa fasilitas kesehatan dapat menangani penerimaan pasien secara efisien dan efektif, meminimalkan waktu tunggu dan mengoptimalkan pemanfaatan sumber daya. Algoritma utama yang digunakan dalam domain ini meliputi (Gambar 14.1):



Gambar 14.1 Algoritma pembelajaran yang digunakan dalam perawatan kesehatan.

Teori Antrian

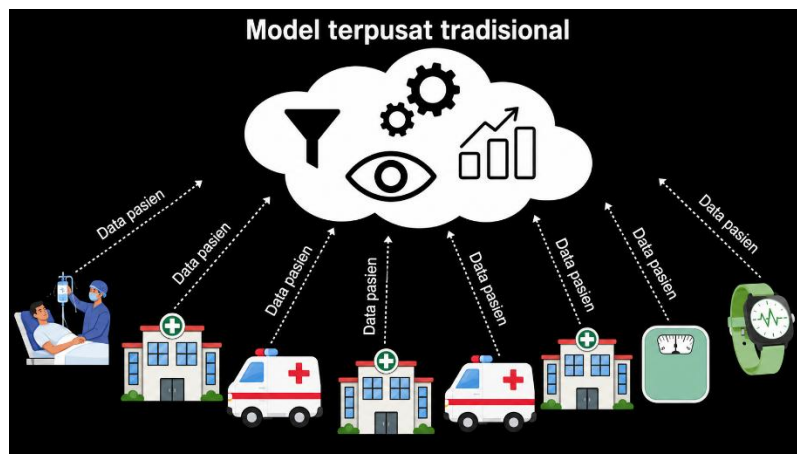
Teori antrian memodelkan kedatangan pasien, proses layanan, dan waktu tunggu, memberikan pendekatan terstruktur untuk memahami dan mengoptimalkan aliran pasien melalui sistem perawatan kesehatan. Dengan menganalisis berbagai parameter seperti tingkat kedatangan, waktu layanan, dan panjang antrian, teori antrian membantu mengidentifikasi potensi hambatan dan inefisiensi dalam penjadwalan pasien. Kerangka analitis ini memungkinkan manajer perawatan kesehatan untuk memprediksi waktu tunggu, mengalokasikan sumber daya secara lebih efektif, dan menerapkan strategi untuk memperlancar aliran pasien. Pada akhirnya, penerapan teori antrian dapat meningkatkan



kepuasan pasien dengan mengurangi waktu tunggu dan meningkatkan efisiensi keseluruhan sistem penyampaian perawatan kesehatan.

Algoritma Simulasi

Algoritma simulasi menciptakan model virtual dari proses perawatan kesehatan, memungkinkan manajer perawatan kesehatan untuk memprediksi dan menganalisis dampak berbagai strategi penjadwalan pada aliran pasien dan pemanfaatan sumber daya. Dengan mensimulasikan skenario dunia nyata, algoritma ini dapat menguji berbagai pendekatan untuk mengelola janji temu pasien, tingkat kepegawaian, dan penggunaan peralatan tanpa mengganggu operasi aktual. Hal ini memungkinkan organisasi layanan kesehatan untuk mengidentifikasi strategi paling efektif untuk meminimalkan waktu tunggu, menyeimbangkan beban kerja, dan mengoptimalkan penggunaan sumber daya (Gambar 14.2). Wawasan yang diperoleh dari model simulasi membantu dalam pengambilan keputusan berbasis data yang meningkatkan efisiensi, meningkatkan hasil pasien, dan memastikan kualitas perawatan yang lebih tinggi.



Gambar 14.2 Alur data dalam perawatan kesehatan.

Pemrograman Linier

Teknik pemrograman linier digunakan untuk mengoptimalkan alokasi sumber daya dalam pengaturan perawatan kesehatan, memastikan bahwa staf dan peralatan dimanfaatkan seefisien mungkin. Dengan merumuskan masalah alokasi sebagai serangkaian persamaan dan pertidaksamaan linier, teknik ini dapat mengidentifikasi cara optimal untuk mendistribusikan sumber daya untuk memenuhi permintaan pasien sambil meminimalkan biaya. Pemrograman linier dapat menangani berbagai kendala dan tujuan, seperti memaksimalkan jumlah pasien, meminimalkan waktu tunggu, dan menyeimbangkan beban kerja di antara staf. Pendekatan ini memungkinkan manajer perawatan kesehatan untuk membuat keputusan yang tepat yang meningkatkan efisiensi operasional, mengurangi pemborosan, dan meningkatkan kualitas perawatan secara keseluruhan yang diberikan kepada pasien.



14.2 TEKNOLOGI PENGEMASAN OBAT

Pengemasan obat dalam industri perawatan kesehatan melibatkan pengemasan obat secara sistematis untuk memastikan keamanan, khasiat, dan kepatuhan terhadap standar peraturan. Algoritma modern memainkan peran penting dalam mengotomatisasi proses pengemasan obat. Algoritma utama meliputi:

Otomasi Proses Robotik

Otomasi proses robotik (RPA) mengotomatiskan tugas-tugas berulang dalam proses pengemasan obat, secara signifikan meningkatkan efisiensi dan mengurangi potensi kesalahan manusia. Dengan menggunakan robot perangkat lunak untuk menangani tugas-tugas seperti pelabelan, penyortiran, dan pengemasan, RPA memastikan akurasi dan kecepatan yang konsisten, yang sangat penting dalam industri farmasi. Sistem otomatis ini dapat bekerja terus menerus tanpa kelelahan, sehingga menghasilkan produktivitas yang lebih tinggi dan biaya operasional yang lebih rendah. Selain itu, integrasi RPA memungkinkan pemantauan dan pengumpulan data secara waktu nyata, memfasilitasi kontrol kualitas yang lebih baik dan kepatuhan terhadap standar peraturan. Implementasi RPA dalam pengemasan obat menyederhanakan operasi, meningkatkan keandalan, dan membebaskan pekerja manusia untuk fokus pada aktivitas yang lebih kompleks dan bernilai tambah.

Pengenalan Karakter Optik

Algoritma pengenalan karakter optik (OCR) digunakan untuk memverifikasi label dan memastikan bahwa informasi yang akurat dicetak pada kemasan obat, sehingga meningkatkan ketertelusuran di seluruh rantai pasokan farmasi. Dengan mengubah teks cetak atau tulisan tangan menjadi data yang dapat dibaca mesin, teknologi OCR dapat dengan cepat dan akurat memeriksa bahwa semua label berisi nama obat, dosis, nomor batch, dan tanggal kedaluwarsa yang benar. Otomatisasi ini tidak hanya mengurangi risiko kesalahan manusia tetapi juga mempercepat proses verifikasi, memastikan kepatuhan terhadap persyaratan peraturan yang ketat. Peningkatan ketertelusuran melalui OCR meningkatkan kemampuan untuk melacak dan memanggil kembali batch obat tertentu jika perlu, sehingga memastikan keselamatan pasien dan menjaga integritas rantai pasokan farmasi.

Analisis Prediktif

Algoritma analitik prediktif memprediksi permintaan dan mengelola tingkat persediaan untuk memastikan bahwa proses pengemasan selaras dengan persyaratan rantai pasokan. Dengan menganalisis data historis, tren pasar, dan faktor relevan lainnya, algoritma ini dapat secara akurat memprediksi permintaan di masa mendatang untuk berbagai obat. Pandangan ke depan ini memungkinkan perusahaan farmasi untuk menyesuaikan jadwal pengemasan dan tingkat persediaan mereka secara proaktif, mencegah kekurangan dan kelebihan stok. Dengan menyelaraskan proses pengemasan dengan dinamika rantai pasokan waktu nyata, analitik prediktif membantu mempertahankan tingkat persediaan yang optimal, mengurangi pemborosan, dan memastikan pengiriman produk tepat waktu. Hal ini meningkatkan efisiensi operasional secara keseluruhan dan memastikan bahwa rantai pasokan tetap responsif terhadap fluktuasi permintaan pasar.



14.3 KEAMANAN DATA PASIEN

Keamanan data merupakan perhatian utama dalam perawatan kesehatan cerdas, di mana informasi pasien yang sensitif terus-menerus dihasilkan, ditransmisikan, dan disimpan. Algoritma kunci yang digunakan untuk memastikan keamanan data meliputi:

Algoritma Enkripsi

Algoritma enkripsi seperti Advanced Encryption Standard (AES) memainkan peran penting dalam melindungi data pasien dengan mengubahnya menjadi format yang tidak dapat dibaca oleh pengguna yang tidak berwenang. AES, protokol enkripsi yang sangat terpercaya, mengubah informasi sensitif seperti rekam medis, detail pribadi, dan riwayat perawatan menjadi ciphertext, yang hanya dapat diuraikan oleh individu yang memiliki kunci dekripsi yang benar. Proses ini memastikan bahwa data pasien tetap aman dari potensi pelanggaran dan serangan siber, melindungi privasi dan menjaga integritas sistem informasi perawatan kesehatan. Menerapkan enkripsi AES membantu penyedia layanan kesehatan mematuhi peraturan perlindungan data yang ketat dan membangun kepercayaan pasien dengan menunjukkan komitmen yang kuat terhadap keamanan data.

Teknologi Blockchain

Teknologi blockchain menawarkan metode terdesentralisasi dan aman untuk menyimpan rekam medis pasien, memastikan integritas dan ketertelusuran data. Dengan mendistribusikan data di seluruh jaringan komputer, blockchain menciptakan buku besar yang tidak dapat diubah di mana setiap transaksi atau entri data dicatat dengan aman dan dihubungkan dengan yang sebelumnya. Arsitektur ini mencegah perubahan yang tidak sah dan menyediakan riwayat yang transparan dan dapat diverifikasi dari semua modifikasi, sehingga meningkatkan kepercayaan pada keakuratan dan keaslian rekam medis pasien. Selain itu, teknik kriptografi blockchain melindungi informasi sensitif, memastikan privasi dan kepatuhan terhadap peraturan perlindungan data. Sifat terdesentralisasi blockchain menghilangkan titik kegagalan tunggal, mengurangi risiko pelanggaran data dan meningkatkan keamanan keseluruhan sistem informasi perawatan kesehatan.

Pembelajaran Mesin untuk Deteksi Anomali

Algoritma pembelajaran mesin berperan penting dalam mendeteksi pola yang tidak biasa yang dapat mengindikasikan pelanggaran keamanan atau akses tidak sah ke data pasien. Dengan terus menganalisis sejumlah besar data dan belajar dari pola historis, algoritma ini dapat mengidentifikasi anomali yang menyimpang dari perilaku normal, seperti upaya login yang tidak biasa, pola akses data yang tidak teratur, atau aktivitas jaringan yang tidak terduga. Ketika anomali tersebut terdeteksi, sistem dapat memicu peringatan, memungkinkan investigasi dan respons cepat terhadap potensi ancaman. Pendekatan proaktif terhadap keamanan ini membantu melindungi informasi pasien yang sensitif, memastikan kepatuhan terhadap persyaratan peraturan dan menjaga integritas dan kerahasiaan data perawatan kesehatan.



14.4 SISTEM PEMANTAUAN NUTRISI OTOMATIS

Sistem pemantauan nutrisi otomatis memanfaatkan algoritma untuk menilai dan melacak asupan makanan dan status gizi individu. Algoritma utama meliputi:

Algoritma Penilaian Diet

Algoritma ini menganalisis data asupan makanan untuk memperkirakan konsumsi nutrisi dan mengidentifikasi kekurangan, memberikan wawasan berharga tentang status gizi individu. Dengan memproses catatan konsumsi makanan yang terperinci, algoritma dapat menguraikan kandungan nutrisi setiap item makanan, menghitung asupan vitamin, mineral, protein, dan nutrisi penting lainnya. Analisis ini membantu mengidentifikasi kesenjangan atau kekurangan dalam diet yang mungkin perlu ditangani untuk menjaga kesehatan optimal. Selain itu, algoritma dapat menawarkan rekomendasi diet yang dipersonalisasi berdasarkan kekurangan yang teridentifikasi, mendorong kebiasaan diet yang lebih baik dan mencegah potensi masalah kesehatan yang terkait dengan gizi buruk.

Sistem Rekomendasi

Sistem rekomendasi menggunakan penyaringan kolaboratif dan penyaringan berbasis konten untuk menawarkan saran diet yang dipersonalisasi yang disesuaikan dengan preferensi dan tujuan kesehatan individu. Penyaringan kolaboratif menganalisis pilihan diet dan umpan balik dari pengguna serupa untuk menyarankan makanan dan rencana makan yang sesuai dengan selera dan kebiasaan individu. Penyaringan berbasis konten, di sisi lain, memeriksa atribut berbagai item makanan, seperti kandungan nutrisi dan bahan-bahan, untuk merekomendasikan pilihan yang memenuhi persyaratan diet dan tujuan kesehatan tertentu. Dengan mengintegrasikan kedua pendekatan ini, sistem rekomendasi dapat memberikan saran diet yang sangat disesuaikan, meningkatkan kepuasan pengguna dan mendukung pola makan yang lebih sehat. Panduan yang dipersonalisasi ini sangat berharga untuk mengelola kondisi kesehatan tertentu, mencapai keseimbangan nutrisi, dan memenuhi target kesehatan pribadi.

Pengenalan Gambar

Algoritma pengenalan gambar mengidentifikasi item makanan dan ukuran porsi dari foto, membantu pengguna dalam mencatat asupan diet mereka secara akurat. Dengan memanfaatkan teknik pembelajaran mesin tingkat lanjut, algoritma ini dapat menganalisis gambar makanan dan camilan untuk mengenali berbagai komponen makanan dan memperkirakan jumlahnya. Teknologi ini menyederhanakan proses pelacakan diet, karena pengguna dapat mengambil foto makanan mereka daripada memasukkan deskripsi rinci secara manual. Algoritma kemudian mengubah informasi visual ini menjadi data nutrisi yang tepat, membantu pengguna memantau konsumsi kalori, makronutrien, dan mikronutrien mereka. Pendekatan otomatis ini meningkatkan akurasi catatan diet dan mendukung individu dalam menjaga pola makan seimbang dan mencapai tujuan kesehatan dan kebugaran mereka.

14.5 LATIHAN

1. Apa itu teori antrian, dan bagaimana penerapannya untuk mengoptimalkan penjadwalan pasien di bidang kesehatan?



2. Bagaimana algoritma simulasi dapat digunakan untuk memprediksi dampak strategi penjadwalan yang berbeda terhadap alur pasien dan pemanfaatan sumber daya? Berikan penjelasan rinci dan contoh model simulasi.
3. Jelaskan peran pemrograman linier dalam mengoptimalkan alokasi sumber daya untuk penjadwalan alur pasien. Buat contoh model pemrograman linier untuk lingkungan perawatan kesehatan.
4. Rancang solusi RPA untuk otomatisasi pengemasan obat di industri perawatan kesehatan. Apa langkah-langkah kunci yang terlibat, dan bagaimana Anda memastikan akurasi dan efisiensi?
5. Kembangkan sistem OCR untuk meningkatkan ketertelusuran dan verifikasi label obat. Jelaskan mekanisme kerjanya dan tantangan utamanya.
6. Buat model analitik prediktif untuk memprediksi permintaan obat dan mengoptimalkan tingkat persediaan dalam kemasan obat.
7. Kembangkan solusi berbasis blockchain untuk mengamankan rekam medis pasien dalam penyimpanan data kesehatan. Diskusikan keunggulan dan tantangan implementasinya.
8. Implementasikan model pembelajaran mesin untuk deteksi anomali dalam data kesehatan. Seberapa efektifkah model ini dalam mengidentifikasi pelanggaran keamanan?
9. Rancang sistem rekomendasi menggunakan penyaringan kolaboratif atau penyaringan berbasis konten untuk memberikan saran diet yang dipersonalisasi.
10. Kembangkan sistem pengenalan gambar untuk mencatat asupan makanan secara akurat. Diskusikan potensi tantangan dan manfaatnya.



BAB 15

ALGORITMA EKOLOGI DAN ENERGI

15.1 URUN DAYA PEMANTAUAN UDARA KOTA

Integrasi algoritma modern dalam infrastruktur lingkungan dan energi telah secara signifikan meningkatkan efisiensi, keandalan, dan keberlanjutan sistem ini. Algoritma canggih memungkinkan pemantauan, pengelolaan, dan optimasi sumber daya yang lebih baik, yang mengarah pada pengambilan keputusan yang lebih tepat dan peningkatan perlindungan lingkungan. Bab ini mengeksplorasi berbagai algoritma modern yang digunakan dalam sektor-sektor utama infrastruktur lingkungan dan energi, termasuk pemantauan polusi udara perkotaan, penjadwalan energi hijau, jaringan pintar, sistem pengelolaan limbah, dan pemantauan drone.

- Agregasi data tingkat lanjut dan crowdsensing untuk pemantauan polusi udara perkotaan
- Teknik optimasi dan peramalan untuk penjadwalan energi hijau
- Manajemen jaringan cerdas yang cerdas: Penyeimbangan beban, deteksi kesalahan, dan optimasi penyimpanan energi
- Sistem pengelolaan limbah cerdas berbasis AI untuk pembangunan perkotaan berkelanjutan
- Pemantauan lingkungan dan infrastruktur berbasis drone menggunakan pembelajaran mendalam dan fusi data

Crowdsensing melibatkan pengumpulan dan analisis data dari sejumlah besar individu menggunakan perangkat seluler dan sensor untuk memantau tingkat polusi udara perkotaan. Pendekatan ini memanfaatkan ketersediaan luas ponsel pintar dan perangkat portabel lainnya untuk mengumpulkan data secara waktu nyata, yang kemudian diproses menggunakan algoritma canggih.

Algoritma

Algoritma yang digunakan dalam Crowdsensing untuk Pemantauan Polusi Udara Perkotaan adalah sebagai berikut:

- **Algoritma Agregasi Data:** Algoritma ini mengagregasi data dari berbagai sumber, memastikan keandalan dan akurasi data dengan menyaring noise dan outlier. Dengan menggabungkan dataset dari berbagai asal, algoritma ini menciptakan pandangan informasi yang komprehensif dan terpadu. Algoritma ini menggunakan teknik canggih untuk mendeteksi dan menghilangkan inkonsistensi, kesalahan, dan titik data yang tidak relevan, sehingga meningkatkan kualitas keseluruhan data yang diagregasi. Proses ini tidak hanya meningkatkan akurasi data tetapi juga meningkatkan keandalannya, sehingga lebih bermanfaat untuk analisis dan pengambilan keputusan. Algoritma agregasi data sangat berharga di bidang-bidang seperti perawatan



kesehatan, keuangan, dan telekomunikasi, di mana data yang akurat dan andal sangat penting untuk efisiensi operasional dan perencanaan strategis.

- **Algoritma Pembelajaran Mesin:** Model pembelajaran mesin, seperti analisis regresi dan jaringan saraf, digunakan untuk memprediksi tingkat polusi dan mengidentifikasi pola dalam data lingkungan. Dengan menganalisis data historis tentang faktor-faktor seperti emisi, kondisi cuaca, dan fitur geografis, model-model ini dapat memprediksi tingkat polusi di masa depan dengan akurasi tinggi. Analisis regresi membantu dalam memahami hubungan antara berbagai variabel dan dampaknya terhadap polusi, sementara jaringan saraf dapat mempelajari pola kompleks dan nonlinier dalam data. Kemampuan prediksi ini memungkinkan para pembuat kebijakan dan lembaga lingkungan untuk menerapkan langkah-langkah proaktif untuk mengurangi polusi dan melindungi kesehatan masyarakat. Selain itu, algoritma pembelajaran mesin dapat mengungkap tren dan anomali tersembunyi dalam data, memberikan wawasan berharga untuk pengkajian dan perencanaan strategis dalam pengelolaan lingkungan.
- **Analisis Geospasial:** Algoritma untuk analisis geospasial, seperti kriging dan interpolasi spasial, memainkan peran penting dalam menciptakan peta polusi terperinci dengan memperkirakan tingkat polusi di lokasi yang tidak dipantau berdasarkan data dari sensor terdekat. Kriging, sebuah metode statistik, menggunakan titik data yang diketahui untuk memprediksi nilai di lokasi yang tidak diketahui, memberikan distribusi spasial polutan yang sangat akurat. Teknik interpolasi spasial, di sisi lain, menghasilkan permukaan kontinu dari titik data diskrit, memastikan cakupan komprehensif di area yang luas. Metode ini memungkinkan visualisasi titik-titik panas polusi dan tren di suatu wilayah geografis, memfasilitasi pemantauan dan pengelolaan lingkungan yang lebih baik. Dengan memanfaatkan analisis geospasial, pihak berwenang dan para ahli dapat memperoleh wawasan yang lebih dalam tentang pola polusi, memungkinkan pengambilan keputusan yang lebih tepat dan intervensi yang ditargetkan untuk meningkatkan kualitas udara dan kesehatan masyarakat.

Contoh Aplikasi

Di daerah perkotaan, sistem crowdsensing mengumpulkan data kualitas udara dari ponsel pintar warga yang dilengkapi sensor. Data tersebut dikumpulkan dan diproses menggunakan algoritma pembelajaran mesin untuk memprediksi titik-titik polusi dan trennya. Analisis geospasial kemudian digunakan untuk membuat peta polusi secara waktu nyata, yang membantu pemerintah kota untuk mengambil tindakan tepat waktu guna mengurangi tingkat polusi.

15.2 MANAJEMEN PERMINTAAN ENERGI HIJAU

Penjadwalan energi hijau berfokus pada pengoptimalan penggunaan sumber energi terbarukan untuk mengelola permintaan pada jaringan energi. Hal ini melibatkan penyeimbangan pasokan dan permintaan energi, mengurangi ketergantungan pada sumber yang tidak terbarukan, dan meningkatkan efisiensi jaringan.



Algoritma

Algoritma yang digunakan dalam penjadwalan energi hijau untuk manajemen sisi permintaan adalah sebagai berikut:

- **Algoritma Optimasi:** Algoritma seperti pemrograman linier, pemrograman bilangan bulat campuran, dan algoritma genetik digunakan untuk mengoptimalkan penjadwalan sumber energi terbarukan berdasarkan prediksi permintaan dan ketersediaan. Teknik pemrograman linier memecahkan alokasi sumber daya terbaik dalam serangkaian kendala, memastikan distribusi energi yang efisien. Pemrograman bilangan bulat campuran meningkatkan hal ini dengan menangani variabel kontinu dan diskrit, sehingga cocok untuk skenario penjadwalan kompleks yang melibatkan beragam sumber energi. Algoritma genetika, yang terinspirasi oleh seleksi alam, secara iteratif mencari solusi optimal dengan mengembangkan populasi jadwal kandidat. Teknik optimasi ini memastikan bahwa energi terbarukan dimanfaatkan dan didistribusikan dengan cara yang paling efisien, menyeimbangkan pasokan dengan permintaan sambil meminimalkan biaya dan memaksimalkan penggunaan sumber daya berkelanjutan. Hal ini mengarah pada sistem energi yang lebih andal dan hemat biaya, yang pada akhirnya mendukung transisi ke jaringan energi yang lebih hijau.
- **Algoritma Peramalan:** Analisis deret waktu dan model pembelajaran mesin merupakan alat penting untuk memprediksi permintaan energi dan produksi energi terbarukan, memfasilitasi penjadwalan yang lebih akurat. Analisis deret waktu memeriksa data historis untuk mengidentifikasi tren, pola musiman, dan perilaku siklik, memungkinkan perkiraan yang tepat tentang kebutuhan energi di masa mendatang. Model pembelajaran mesin, seperti jaringan saraf dan algoritma regresi, meningkatkan prediksi ini dengan belajar dari kumpulan data yang luas dan mengidentifikasi hubungan kompleks antara variabel, seperti kondisi cuaca, pola konsumsi, dan tingkat pembangkitan energi. Dengan mengintegrasikan teknik peramalan ini, penyedia energi dapat lebih baik mencocokkan pasokan dengan permintaan, mengoptimalkan penggunaan sumber daya terbarukan dan meningkatkan stabilitas jaringan. Pendekatan proaktif ini membantu meminimalkan pemborosan energi, mengurangi biaya, dan memastikan pasokan energi bersih yang andal.
- **Algoritma Respons Permintaan:** Algoritma respons permintaan memainkan peran penting dalam menyesuaikan konsumsi energi berdasarkan kondisi jaringan waktu nyata, memberikan insentif bagi pengguna untuk menggeser penggunaan energi mereka ke waktu di luar jam sibuk. Algoritma ini menganalisis beban jaringan saat ini dan yang diprediksi, harga energi, dan pola konsumsi pengguna untuk mengelola permintaan secara dinamis. Dengan memberi sinyal kepada pengguna atau sistem otomatis untuk mengurangi atau menggeser penggunaan energi mereka selama periode puncak, algoritma ini membantu menyeimbangkan beban pada jaringan, mencegah kelebihan beban dan mengurangi kebutuhan akan pembangkit listrik puncak yang mahal dan berpolusi. Insentif seperti tarif energi yang lebih rendah atau imbalan finansial langsung mendorong pengguna untuk berpartisipasi dalam program



respons permintaan. Hal ini tidak hanya meningkatkan keandalan dan efisiensi jaringan listrik tetapi juga mendorong kebiasaan konsumsi energi yang lebih berkelanjutan dan mendukung integrasi sumber energi terbarukan.

Contoh Aplikasi

Sebuah perusahaan energi menggunakan algoritma optimasi untuk menjadwalkan penggunaan sumber energi surya dan angin berdasarkan prakiraan cuaca dan data permintaan historis. Algoritma peramalan memprediksi permintaan energi dan produksi energi terbarukan untuk hari berikutnya, memungkinkan perusahaan untuk merencanakan campuran sumber energi yang optimal. Algoritma respons permintaan kemudian menyesuaikan pola penggunaan energi konsumen agar selaras dengan ketersediaan energi terbarukan.

15.3 JARINGAN CERDAS

Jaringan cerdas mengintegrasikan teknologi digital dengan jaringan listrik tradisional untuk meningkatkan pemantauan, pengendalian, dan pengelolaan distribusi listrik. Hal ini menghasilkan peningkatan keandalan, efisiensi, dan keberlanjutan pasokan listrik. **Algoritma Utama – Algoritma yang digunakan dalam jaringan cerdas adalah sebagai berikut:**

- **Algoritma Optimasi Jaringan:** Algoritma optimasi jaringan mencakup berbagai teknik untuk penyeimbangan beban, pengendalian tegangan, dan konfigurasi ulang jaringan untuk meningkatkan kinerja jaringan. Algoritma penyeimbangan beban mendistribusikan permintaan listrik secara merata di seluruh jaringan, mencegah kelebihan beban dan memastikan penggunaan sumber daya yang tersedia secara efisien. Algoritma pengendalian tegangan mempertahankan tingkat tegangan optimal di seluruh jaringan, yang sangat penting untuk stabilitas dan keandalan pengiriman daya. Algoritma konfigurasi ulang jaringan secara dinamis menyesuaikan konfigurasi jaringan sebagai respons terhadap perubahan kondisi, seperti gangguan atau pola permintaan yang bervariasi, untuk meminimalkan kerugian dan meningkatkan ketahanan. Dengan mengintegrasikan teknik optimasi ini, operator jaringan dapat memastikan pasokan listrik yang lebih andal, efisien, dan berkelanjutan, mengakomodasi kompleksitas dan variabilitas yang semakin meningkat yang terkait dengan sumber energi terbarukan.
- **Deteksi dan Diagnosis Gangguan:** Algoritma pembelajaran mesin dan pengenalan pola sangat penting dalam mendeteksi dan mendiagnosis gangguan dalam jaringan, memastikan respons yang cepat dan meminimalkan waktu henti. Algoritma ini menganalisis sejumlah besar data dari sensor dan perangkat pemantauan untuk mengidentifikasi anomali dan pola yang menunjukkan potensi gangguan. Dengan memanfaatkan teknik canggih seperti jaringan saraf dan pohon keputusan, algoritma dapat secara akurat menentukan lokasi dan sifat masalah, mulai dari kerusakan peralatan hingga kegagalan saluran. Deteksi gangguan yang cepat dan tepat ini memungkinkan operator jaringan untuk mengambil tindakan korektif segera, mengurangi durasi dan dampak pemadaman. Selain itu, kemampuan pembelajaran



berkelanjutan dari algoritma ini meningkatkan akurasi dari waktu ke waktu, berkontribusi pada keandalan dan ketahanan keseluruhan jaringan listrik.

- **Manajemen Penyimpanan Energi:** Algoritma yang dirancang untuk manajemen penyimpanan energi mengoptimalkan penggunaan sistem penyimpanan energi untuk menyeimbangkan pasokan dan permintaan secara efektif. Algoritma ini secara cerdas mengontrol siklus pengisian dan pengosongan perangkat penyimpanan, seperti baterai, dengan menganalisis kondisi jaringan secara waktu nyata, harga energi, dan perkiraan permintaan. Selama periode permintaan rendah dan produksi energi berlebih, seperti dari sumber terbarukan, algoritma ini mengarahkan sistem penyimpanan untuk menyimpan energi berlebih. Sebaliknya, selama waktu permintaan puncak, algoritma ini memastikan bahwa energi yang tersimpan dilepaskan kembali ke jaringan secara efisien. Manajemen strategis ini membantu menstabilkan jaringan, mengurangi ketergantungan pada pembangkit listrik puncak, dan menurunkan biaya energi. Dengan memaksimalkan pemanfaatan energi yang tersimpan, algoritma ini juga meningkatkan integrasi sumber energi terbarukan yang bersifat intermiten, berkontribusi pada infrastruktur energi yang lebih berkelanjutan dan tangguh.

Contoh Aplikasi

Sebuah perusahaan utilitas menerapkan sistem jaringan cerdas dengan algoritma optimasi jaringan untuk menyeimbangkan beban dan menjaga tingkat tegangan di seluruh jaringan. Algoritma deteksi kesalahan memantau jaringan untuk setiap anomali, memungkinkan diagnosis dan perbaikan masalah dengan cepat. Algoritma manajemen penyimpanan energi mengoptimalkan pengisian dan pengosongan baterai untuk memastikan pasokan listrik yang stabil bahkan selama periode permintaan puncak.

15.4 SISTEM MANAJEMEN LIMBAH CERDAS

Sistem manajemen limbah cerdas menggunakan sensor, analitik data, dan algoritma untuk mengoptimalkan pengumpulan dan pembuangan limbah. Sistem ini meningkatkan efisiensi operasional, mengurangi biaya, dan meminimalkan dampak lingkungan. **Algoritma Utama – Algoritma yang digunakan dalam sistem manajemen limbah cerdas adalah sebagai berikut:**

- **IoT dan Jaringan Sensor:** Algoritma seperti Algoritma Perutean Dinamis mengelola data dari sensor IoT yang ditempatkan di tempat sampah untuk memantau tingkat pengisian dan mengoptimalkan rute pengumpulan. Sensor-sensor ini terus menerus mengirimkan data mengenai status pengisian setiap tempat sampah ke sistem manajemen pusat. Algoritma Perutean Dinamis memproses data waktu nyata ini, menganalisis tingkat pengisian untuk menentukan tempat sampah mana yang memerlukan pengumpulan segera dan mana yang dapat menunggu. Dengan menghitung rute pengumpulan yang paling efisien berdasarkan tingkat pengisian saat ini dan lokasi tempat sampah, algoritma ini meminimalkan jarak tempuh dan konsumsi bahan bakar untuk kendaraan pengumpul sampah. Hal ini tidak hanya mengurangi



biaya operasional dan dampak lingkungan tetapi juga memastikan pembuangan sampah tepat waktu, mencegah luapan dan menjaga kebersihan perkotaan. Mengintegrasikan algoritma cerdas tersebut ke dalam sistem pengelolaan sampah meningkatkan efisiensi, keberlanjutan, dan respons terhadap perubahan kondisi.

- **Algoritma Optimasi Rute:** Algoritma seperti masalah pedagang keliling (TSP) dan masalah perutean kendaraan (VRP) mengoptimalkan rute pengumpulan sampah, mengurangi waktu tempuh dan konsumsi bahan bakar.
- **Analisis Prediktif:** Model pembelajaran mesin, seperti Algoritma Random Forest, memprediksi pola produksi limbah, sehingga memungkinkan perencanaan dan alokasi sumber daya yang lebih baik. Algoritma Random Forest, sebuah metode pembelajaran ensemble, menganalisis data historis tentang produksi limbah, dengan menggabungkan variabel seperti waktu dalam setahun, kepadatan penduduk, dan aktivitas ekonomi. Dengan pelatihan pada kumpulan data yang luas ini, algoritma belajar untuk mengidentifikasi pola dan tren kompleks dalam produksi limbah. Setelah dilatih, algoritma dapat secara akurat memprediksi volume limbah di masa mendatang, memberikan wawasan berharga bagi perencana kota dan perusahaan pengelolaan limbah. Kemampuan prediktif ini memungkinkan penjadwalan pengumpulan limbah yang optimal, alokasi sumber daya yang efisien, dan pengelolaan fasilitas pembuangan limbah yang proaktif. Dengan mengantisipasi fluktuasi produksi limbah, kota dapat mengurangi biaya, meningkatkan keandalan layanan, dan meminimalkan dampak lingkungan.

Contoh Aplikasi

Sebuah kota menerapkan sistem pengelolaan limbah cerdas dengan sensor IoT di tempat sampah untuk memantau tingkat pengisian. Algoritma optimasi rute menghitung rute yang paling efisien untuk truk pengumpul limbah, mengurangi biaya operasional dan dampak lingkungan. Model analitik prediktif menganalisis data historis untuk memprediksi tren produksi limbah, sehingga memungkinkan kota untuk mengalokasikan sumber daya secara lebih efektif.

15.5 PEMANTAUAN DRONE

Drone yang dilengkapi dengan sensor dan kamera digunakan untuk memantau kondisi lingkungan, infrastruktur, dan lahan pertanian. Algoritma memproses data yang dikumpulkan oleh drone untuk memberikan wawasan yang dapat ditindaklanjuti. **Algoritma Utama – Algoritma yang digunakan dalam pemantauan drone adalah sebagai berikut:**

- **Algoritma Pemrosesan Gambar:** Jaringan saraf konvolusional (CNN) digunakan untuk menganalisis gambar yang diambil oleh drone, mendeteksi perubahan, anomali, atau fitur spesifik di lingkungan yang dipantau. CNN, jenis algoritma pembelajaran mendalam, unggul dalam memproses dan menafsirkan data visual karena kemampuannya untuk mempelajari dan mengenali pola dan hierarki spasial dalam gambar. Ketika drone mengambil gambar area yang dipantau, CNN memproses gambar-gambar ini untuk mengidentifikasi perubahan signifikan dari waktu ke waktu,



seperti perubahan penggunaan lahan, penggundulan hutan, atau pengembangan infrastruktur. Selain itu, CNN dapat mendeteksi anomali spesifik seperti konstruksi ilegal, bahaya lingkungan, atau variasi kesehatan vegetasi. Dengan terus menganalisis citra drone beresolusi tinggi, CNN menyediakan pemantauan waktu nyata yang akurat untuk area yang luas dan terpencil, meningkatkan kesadaran situasional dan memungkinkan intervensi tepat waktu. Teknologi ini sangat penting untuk aplikasi di bidang pertanian, konservasi lingkungan, manajemen bencana, dan perencanaan kota, di mana pemantauan yang tepat dan efisien sangat penting.

- **Pembelajaran Mesin untuk Deteksi Objek:** Model seperti algoritma You Only Look Once (YOLO) digunakan untuk mengidentifikasi dan mengklasifikasikan objek dalam rekaman drone, memungkinkan tugas-tugas seperti mendeteksi infrastruktur yang rusak dan memantau kesehatan tanaman. YOLO adalah sistem deteksi objek waktu nyata yang menerapkan satu jaringan saraf ke seluruh gambar, membaginya menjadi beberapa wilayah dan memprediksi kotak pembatas dan probabilitas untuk setiap wilayah. Pendekatan ini memungkinkan YOLO untuk mencapai kecepatan dan akurasi deteksi yang tinggi, sehingga cocok untuk memproses volume data besar yang ditangkap oleh drone. Dalam konteks infrastruktur, YOLO dapat dengan cepat mengidentifikasi dan mengkategorikan berbagai jenis kerusakan, seperti retakan pada jembatan atau bangunan yang runtuh, memfasilitasi respons cepat dan upaya perbaikan. Dalam bidang pertanian, YOLO dapat memantau kesehatan tanaman dengan mendeteksi tanda-tanda penyakit, serangan hama, atau kekurangan nutrisi, sehingga memungkinkan intervensi tepat waktu untuk menjaga kondisi tanaman tetap optimal. Kemampuan deteksi objek tingkat lanjut ini meningkatkan efisiensi dan efektivitas pemantauan dan pengelolaan lingkungan yang beragam.
- **Algoritma Fusi Data:** Algoritma fusi data, seperti Filter Kalman, digunakan untuk menggabungkan data dari beberapa sensor pada drone, memberikan pandangan komprehensif tentang area yang dipantau. Filter Kalman adalah algoritma estimasi optimal yang mengintegrasikan informasi dari berbagai sensor, seperti kamera, LiDAR, dan sensor inframerah, untuk menghasilkan representasi tunggal dan akurat dari lingkungan. Dengan terus memperbarui estimasi berdasarkan data sensor dan memperhitungkan ketidakpastian dan noise, Filter Kalman meningkatkan keandalan dan akurasi keluaran gabungan. Integrasi data yang komprehensif ini sangat penting untuk aplikasi yang membutuhkan kesadaran situasional yang tepat, seperti pemantauan lingkungan, penanggulangan bencana, dan pengelolaan pertanian. Data yang digabungkan memungkinkan deteksi perubahan halus, identifikasi anomali, dan pembuatan peta terperinci, yang pada akhirnya mengarah pada pengambilan keputusan yang lebih tepat dan pengelolaan area yang dipantau yang lebih efektif.

Contoh Aplikasi

Sebuah perusahaan pertanian menggunakan drone yang dilengkapi dengan kamera multispektral untuk memantau kesehatan tanaman. Algoritma pemrosesan gambar menganalisis gambar yang diambil untuk mendeteksi tanda-tanda penyakit atau kekurangan



nutrisi. Model pembelajaran mesin mengklasifikasikan status kesehatan berbagai area tanaman, dan algoritma fusi data menggabungkan temuan ini dengan data cuaca untuk memberikan wawasan yang dapat ditindaklanjuti kepada petani untuk intervensi yang ditargetkan.

15.6 LATIHAN

Algoritma agregasi data untuk menggabungkan dan membersihkan data dari berbagai sumber. (a) Model pembelajaran mesin untuk memprediksi tingkat polusi dan mengidentifikasi pola. (b) Analisis geospasial untuk membuat peta polusi terperinci secara waktu nyata. (c) Diskusi tentang bagaimana mengatasi tantangan yang terkait dengan keragaman, keandalan, dan skalabilitas data. Mengembangkan sistem penjadwalan energi hijau yang mengoptimalkan penggunaan sumber energi terbarukan untuk mengelola permintaan pada jaringan energi. Proyek Anda harus mencakup: (a) Algoritma optimasi untuk menjadwalkan energi terbarukan berdasarkan permintaan yang diprediksi. (b) Algoritma peramalan untuk memprediksi permintaan dan produksi energi. (c) Algoritma respons permintaan untuk menyesuaikan konsumsi energi secara waktu nyata. (d) Studi kasus yang menunjukkan bagaimana sistem Anda meningkatkan efisiensi jaringan dan mengurangi ketergantungan pada energi non-terbarukan.

1. Algoritma optimasi jaringan untuk penyeimbangan beban, kontrol tegangan, dan konfigurasi ulang jaringan. (a) Algoritma pembelajaran mesin untuk deteksi dan diagnosis kesalahan. (b) Algoritma manajemen penyimpanan energi untuk mengoptimalkan penggunaan sistem penyimpanan energi. (c) Analisis tentang bagaimana sistem Anda meningkatkan keandalan, efisiensi, dan keberlanjutan jaringan.
2. Rancang sistem pengelolaan limbah cerdas yang memanfaatkan sensor IoT dan analitik data. Proyek Anda harus mencakup: (a) Jaringan IoT dan sensor untuk memantau tingkat pengisian tempat sampah. (b) Algoritma optimasi rute untuk menghitung rute pengumpulan limbah yang efisien. (c) Analitik prediktif untuk memperkirakan pola pembangkitan limbah. (d) Diskusi tentang bagaimana sistem Anda mengurangi biaya operasional dan dampak lingkungan. Pemantauan Drone.
3. Kembangkan sistem pemantauan berbasis drone untuk kondisi lingkungan, infrastruktur, atau lahan pertanian. Proyek Anda harus mencakup: (a) Algoritma pemrosesan gambar menggunakan CNN untuk menganalisis gambar yang diambil drone. (b) Model pembelajaran mesin seperti YOLO untuk deteksi dan klasifikasi objek. (c) Algoritma fusi data untuk menggabungkan data dari beberapa sensor drone. (d) Studi kasus yang menggambarkan bagaimana sistem Anda memberikan wawasan yang dapat ditindaklanjuti untuk intervensi yang ditargetkan.



BAB 16

PERTANIAN CERDAS

16.1 PERTANIAN PRESISI

Pertanian cerdas mengintegrasikan teknologi canggih seperti Internet of Things (IoT), kecerdasan buatan (AI), pembelajaran mesin (ML), dan analitik data untuk meningkatkan efisiensi, produktivitas, dan keberlanjutan pertanian. Teknologi ini memungkinkan pemantauan dan pengelolaan proses pertanian secara waktu nyata, mengoptimalkan penggunaan sumber daya, dan meningkatkan hasil panen. Bab ini berfokus pada algoritma kunci yang digunakan dalam pertanian cerdas dan aplikasinya. Tujuan bab ini adalah: Meninjau algoritma kunci yang digunakan dalam pertanian presisi untuk mengoptimalkan produktivitas pertanian.

- Menelaah berbagai algoritma untuk pemantauan kesehatan tanah dan aplikasinya dalam pertanian berkelanjutan.
- Menganalisis berbagai pendekatan algoritmik untuk pengelolaan irigasi dan optimasi sumber daya air.
- Menjelajahi algoritma yang ada untuk prediksi hasil panen dan efektivitasnya dalam memprediksi produksi.
- Memeriksa peran algoritma dalam pengelolaan hama dan penyakit untuk deteksi dan pengendalian dini.

Pertanian presisi, juga dikenal sebagai agrikultur presisi, adalah pendekatan canggih untuk pertanian modern yang memanfaatkan teknologi mutakhir seperti IoT, AI, ML, dan analitik data untuk meningkatkan efisiensi, produktivitas, dan keberlanjutan pertanian secara keseluruhan. Dengan mengintegrasikan teknologi ini, petani dapat mencapai pengambilan keputusan berbasis data, yang mengarah pada peningkatan pengelolaan sumber daya, hasil panen yang lebih tinggi, dan pengurangan dampak lingkungan. Dengan mengintegrasikan algoritma Smart World, pertanian presisi memungkinkan pemantauan dan pengelolaan proses pertanian secara waktu nyata, yang mengarah pada optimalisasi pemanfaatan sumber daya. Hal ini tidak hanya meningkatkan hasil panen tetapi juga meningkatkan keberlanjutan dengan mengurangi penggunaan air yang berlebihan, meminimalkan masukan kimia, dan menurunkan biaya pertanian secara keseluruhan.

Pada akhirnya, pertanian presisi mewakili pergeseran transformatif dalam pertanian, memastikan ketahanan pangan dan mempromosikan konservasi lingkungan dalam menghadapi tantangan global seperti perubahan iklim dan pertumbuhan penduduk. Bagian ini memberikan survei tentang algoritma utama yang digunakan dalam pertanian presisi, pemantauan kesehatan tanah, pengelolaan irigasi, prediksi hasil panen, dan pengelolaan hama dan penyakit. Dengan menganalisis pendekatan dan metodologi yang ada, karya ini bertujuan untuk menyoroti kemajuan dan tantangan terkini dalam pertanian cerdas.



Algoritma Utama

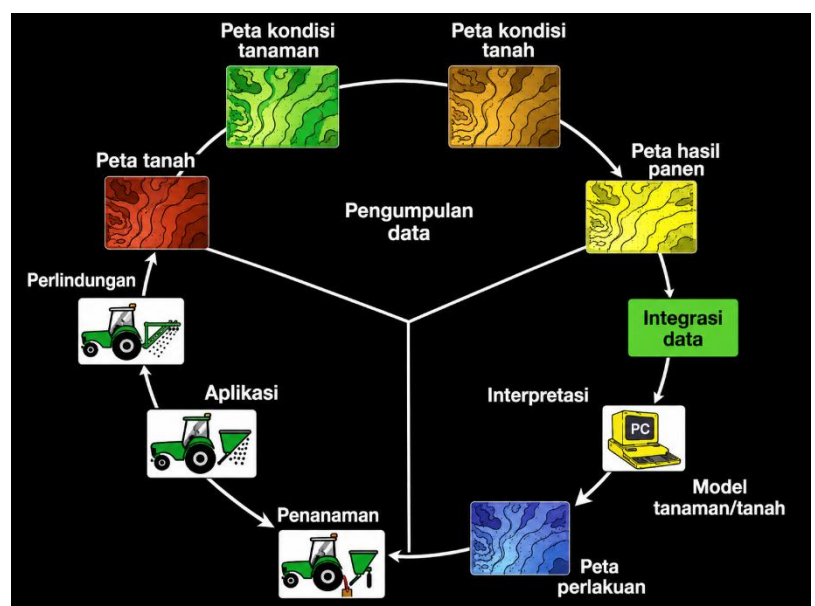
Pertanian presisi, juga dikenal sebagai pertanian akurat, memanfaatkan berbagai algoritma untuk meningkatkan praktik pertanian. Beberapa algoritma utama meliputi:

- **Algoritma Pembelajaran Mesin:** Algoritma ini digunakan untuk analitik prediktif dan pengambilan keputusan. Teknik umum meliputi pembelajaran terawasi (misalnya, regresi, klasifikasi), pembelajaran tak terawasi (misalnya, pengelompokan), dan pembelajaran penguatan.
- **Algoritma Pemrosesan Citra:** Digunakan untuk menganalisis citra satelit dan drone untuk memantau kesehatan tanaman, mendeteksi penyakit, dan memperkirakan hasil panen. Tekniknya meliputi segmentasi citra, deteksi objek, dan klasifikasi.
- **Algoritma Geospasial:** Digunakan untuk memproses data spasial dan menghasilkan peta detail lahan pertanian. Algoritma ini membantu dalam memahami variabilitas tanah, mengoptimalkan pola tanam, dan mengelola sumber daya secara efisien.
- **Algoritma Optimasi:** Digunakan untuk mengoptimalkan berbagai aspek pertanian, seperti penjadwalan irigasi, aplikasi pupuk, dan pengendalian hama. Teknik yang digunakan meliputi pemrograman linier, algoritma genetika, dan metode heuristik.

Contoh Aplikasi

Contoh aplikasi pertanian presisi adalah penggunaan drone yang dilengkapi dengan kamera multispektral untuk memantau kesehatan tanaman. Prosesnya dapat diuraikan sebagai berikut:

- Pengumpulan Data:** Drone terbang di atas lahan pertanian, mengambil gambar multispektral yang mencakup panjang gelombang tampak dan NIR, seperti yang ditunjukkan pada Gambar 16.1.
- Pengolahan Citra:** Citra yang diambil diproses menggunakan algoritma pengolahan citra untuk membuat indeks vegetasi seperti indeks vegetasi perbedaan normalisasi (NDVI). Indeks ini membantu dalam menilai kesehatan dan kekuatan tanaman.
- Analisis Data:** Algoritma ML menganalisis data yang diproses untuk mendeteksi pola dan anomali. Misalnya, mereka dapat mengidentifikasi area yang terkena hama, penyakit, atau kekurangan nutrisi.
- Pengambilan Keputusan:** Berdasarkan analisis,



Gambar 16.1 Pengumpulan data di tanah.

algoritma geospasial menghasilkan peta yang tepat yang menunjukkan berbagai kondisi di seluruh lahan. Algoritma optimasi kemudian digunakan untuk menciptakan



wawasan yang dapat ditindaklanjuti, seperti penyemprotan pestisida yang ditargetkan atau aplikasi pupuk dengan laju variabel.

- e. **Implementasi:** Petani menerapkan tindakan yang direkomendasikan, sehingga meningkatkan hasil panen, mengurangi biaya masukan, dan meminimalkan dampak lingkungan.

16.2 PEMANTAUAN KESEHATAN TANAH

Pemantauan kesehatan tanah melibatkan penilaian dan pengelolaan kualitas tanah untuk memastikan fungsi optimalnya bagi pertumbuhan tanaman, penyaringan air, dan mendukung keanekaragaman hayati. Proses ini biasanya melibatkan kombinasi indikator fisik, kimia, dan biologis untuk mengukur kesehatan tanah. Algoritma utama meliputi:

Algoritma Utama

Pemantauan kesehatan tanah menggunakan berbagai algoritma untuk menilai dan menjaga kualitas tanah. Algoritma utama meliputi:

- **Algoritma Pengambilan Sampel Tanah:** Algoritma ini mengoptimalkan pengumpulan sampel tanah di seluruh lahan. Teknik seperti pengambilan sampel grid, pengambilan sampel zona, dan pengambilan sampel transek umumnya digunakan.
- **Algoritma Analisis Spektral:** Digunakan untuk menganalisis sampel tanah menggunakan data spektral. Algoritma memproses data dari spektroskopi tampak, inframerah dekat (NIR), dan inframerah tengah (MIR) untuk menentukan sifat tanah seperti bahan organik, kadar air, dan kadar nutrisi.
- **Algoritma Pembelajaran Mesin:** Digunakan untuk pemodelan prediktif dan klasifikasi indikator kesehatan tanah. Teknik umum meliputi pohon keputusan, mesin vektor pendukung (SVM), dan jaringan saraf.
- **Algoritma Geostatistik:** Digunakan untuk analisis spasial dan pemetaan sifat tanah. Tekniknya meliputi kriging, co-kriging, dan analisis variogram, yang membantu dalam memahami variabilitas spasial dan membuat peta kesehatan tanah.
- **Algoritma Penggabungan Data:** Algoritma ini mengintegrasikan data dari berbagai sumber, seperti citra satelit, sensor tanah, dan data cuaca, untuk memberikan penilaian komprehensif tentang kesehatan tanah.

Contoh Aplikasi

Contoh aplikasi pemantauan kesehatan tanah melibatkan penggunaan sensor tanah portabel dan analisis geostatistik. Prosesnya dapat diuraikan sebagai berikut:

- a. **Pengumpulan Data:** Sensor tanah portabel digunakan untuk mengumpulkan data waktu nyata tentang sifat tanah seperti pH, konduktivitas listrik, dan kadar air di berbagai lokasi di dalam suatu lahan.
- b. **Pengolahan Data:** Algoritma analisis spektral memproses data sensor untuk menentukan indikator kesehatan tanah utama. Data ini kemudian digunakan untuk mengkalibrasi model ML yang memprediksi tingkat nutrisi tanah dan kandungan bahan organik.



- c. **Analisis Spasial:** Algoritma geostatistik menganalisis data yang diproses untuk memahami variabilitas spasial sifat tanah di seluruh lahan. Teknik Kriging digunakan untuk membuat peta kesehatan tanah yang detail.
- d. **Pengambilan Keputusan:** Berdasarkan analisis, algoritma fusi data mengintegrasikan peta kesehatan tanah dengan sumber data tambahan, seperti citra satelit dan prakiraan cuaca. Integrasi ini membantu dalam pengambilan keputusan yang tepat mengenai praktik pengelolaan tanah.
- e. **Implementasi:** Petani menerapkan praktik pengelolaan tanah yang direkomendasikan, seperti pemupukan dengan laju variabel dan penambahan bahan organik yang ditargetkan, untuk meningkatkan kesehatan tanah dan mengoptimalkan produksi tanaman.

16.3 MANAJEMEN IRIGASI

Manajemen irigasi adalah proses mengendalikan jumlah, waktu, dan metode pemberian air pada tanaman untuk mengoptimalkan pertumbuhan dan memaksimalkan efisiensi penggunaan air. Manajemen irigasi yang efektif sangat penting untuk pertanian berkelanjutan, konservasi air, dan memastikan hasil panen yang tinggi. Manajemen irigasi menggunakan berbagai algoritma untuk mengoptimalkan penggunaan air dan meningkatkan hasil panen. Algoritma utama meliputi: Algoritma Utama

- **Algoritma Kelembaban Tanah:** Algoritma ini mengukur dan memprediksi tingkat kelembaban tanah menggunakan data dari sensor kelembaban tanah. Tekniknya meliputi reflektometri domain waktu (TDR) dan sensor kapasitansi.
- **Algoritma Peramalan Cuaca:** Memanfaatkan data meteorologi untuk memprediksi kondisi cuaca, membantu dalam perencanaan irigasi. Tekniknya meliputi model prediksi cuaca numerik dan peramalan berbasis ML.
- **Algoritma Evapotranspirasi (ET):** Menghitung air yang hilang melalui penguapan dan transpirasi tanaman. Tekniknya meliputi persamaan Penman–Monteith dan metode penginderaan jauh.
- **Sistem Pendukung Keputusan (DSS):** Mengintegrasikan data dari berbagai sumber untuk memberikan wawasan yang dapat ditindaklanjuti untuk penjadwalan irigasi. DSS menggunakan algoritma untuk penggabungan data, analisis, dan optimasi.
- **Algoritma Optimasi:** Digunakan untuk menentukan jadwal dan jumlah irigasi yang paling efisien. Tekniknya meliputi pemrograman linier, algoritma genetika, dan metode heuristik.

Contoh Aplikasi

Contoh manajemen irigasi melibatkan penggunaan sensor kelembaban tanah dan DSS:

- a. **Pengumpulan Data:** Sensor kelembaban tanah ditempatkan di lapangan untuk mengumpulkan data waktu nyata tentang tingkat kelembaban tanah.
- b. **Pengolahan Data:** Algoritma kelembaban tanah memproses data sensor untuk menentukan tingkat kelembaban saat ini dan memprediksi ketersediaan kelembaban di masa mendatang.



- c. **Integrasi Prakiraan Cuaca:** Algoritma prakiraan cuaca memprediksi kondisi cuaca yang akan datang, yang diintegrasikan ke dalam proses perencanaan irigasi.
- d. **Perhitungan ET:** Algoritma ET menghitung kehilangan air melalui penguapan dan transpirasi, memberikan data penting untuk kebutuhan irigasi.
- e. **Pengambilan Keputusan:** DSS menganalisis data yang diproses dan memberikan jadwal dan jumlah irigasi yang optimal berdasarkan kelembaban tanah, prakiraan cuaca, dan perhitungan ET.
- f. **Implementasi:** Petani menerapkan jadwal irigasi yang direkomendasikan, yang mengarah pada penggunaan air yang efisien dan peningkatan hasil panen.

16.4 PREDIKSI HASIL PANEN

Prediksi hasil panen merupakan aspek penting dalam pertanian modern, yang memungkinkan petani, ahli agronomi, dan pembuat kebijakan untuk mengambil keputusan yang tepat mengenai produksi pangan, alokasi sumber daya, dan strategi pasar. Dengan memanfaatkan teknologi canggih seperti pembelajaran mesin, penginderaan jauh, analisis geospasial, dan peramalan deret waktu, model prediksi hasil panen menganalisis berbagai faktor yang memengaruhi produktivitas pertanian. Faktor-faktor ini meliputi pola cuaca, kondisi tanah, kesehatan tanaman, data hasil panen historis, dan citra satelit atau drone secara waktu nyata. Algoritma ML, seperti model regresi, SVM, dan jaringan saraf, banyak digunakan untuk membangun hubungan prediktif antara variabel lingkungan dan hasil panen. Teknik penginderaan jauh, termasuk indeks vegetasi seperti NDVI, memberikan wawasan penting tentang kesehatan tanaman dan tahapan pertumbuhan. Selain itu, algoritma geostatistik membantu dalam analisis variabilitas spasial, sementara teknik fusi data mengintegrasikan berbagai sumber informasi untuk meningkatkan akurasi prediksi. Prediksi hasil panen yang efektif meningkatkan perencanaan pertanian, mengoptimalkan pemanfaatan sumber daya, mengurangi risiko yang terkait dengan variabilitas iklim, dan berkontribusi pada ketahanan pangan global. Dengan semakin tersedianya data beresolusi tinggi dan daya komputasi, model prediktif terus berkembang, menjadikan peramalan hasil panen lebih tepat dan dapat ditindaklanjuti bagi para pemangku kepentingan di sektor pertanian.

Algoritma Utama

Prediksi hasil panen menggunakan berbagai algoritma untuk memprediksi produksi tanaman. Algoritma utama meliputi:

- **Algoritma Pembelajaran Mesin:** Algoritma ini digunakan untuk pemodelan prediktif. Tekniknya meliputi model regresi, SVM, random forests, dan jaringan saraf.
- **Algoritma Analisis Deret Waktu:** Digunakan untuk menganalisis data temporal dan memprediksi hasil panen di masa mendatang. Tekniknya meliputi ARIMA, dekomposisi musiman, dan penghalusan eksponensial.
- **Algoritma Penginderaan Jauh:** Digunakan untuk menganalisis citra satelit dan drone untuk menilai kesehatan tanaman dan memprediksi hasil panen. Tekniknya meliputi klasifikasi citra, indeks vegetasi seperti NDVI, dan deteksi objek.



- **Algoritma Geostatistik:** Algoritma ini menganalisis data spasial untuk memahami variabilitas di berbagai lahan. Teknik yang digunakan meliputi kriging, co-kriging, dan regresi spasial.
- **Algoritma Penggabungan Data:** Mengintegrasikan data dari berbagai sumber, seperti data cuaca, data tanah, dan citra penginderaan jauh, untuk meningkatkan akurasi prediksi.

Contoh Aplikasi

Contoh prediksi hasil panen melibatkan integrasi ML dan data penginderaan jauh:

- a. **Pengumpulan Data:** Mengumpulkan data hasil panen historis, data cuaca, data tanah, dan citra penginderaan jauh untuk lahan pertanian.
- b. **Pengolahan Data:** Menggunakan algoritma penginderaan jauh untuk menganalisis citra satelit dan mengekstrak indeks vegetasi seperti NDVI.
- c. **Pemodelan Prediktif:** Algoritma ML dilatih pada data yang dikumpulkan untuk membangun model prediktif. Model ini menggunakan fitur yang berasal dari data cuaca, tanah, dan penginderaan jauh untuk memprediksi hasil panen.
- d. **Analisis Spasial:** Algoritma geostatistik digunakan untuk menganalisis variabilitas spasial dan menyempurnakan prediksi hasil panen di berbagai lokasi di dalam lahan pertanian.
- e. **Implementasi:** Model prediktif memberikan perkiraan hasil panen, yang membantu petani dalam proses pengambilan keputusan seperti perencanaan panen, optimalisasi penggunaan masukan, dan pengelolaan risiko.

16.5 LATIHAN

1. Evaluasi efektivitas algoritma ML dalam memprediksi kesehatan tanaman dan mengoptimalkan alokasi sumber daya dalam pertanian presisi.
2. Analisis tantangan yang terkait dengan implementasi algoritma pengolahan citra untuk deteksi penyakit secara waktu nyata menggunakan citra drone.
3. Periksa peran algoritma geospasial dalam mengoptimalkan pola tanam dan pengelolaan sumber daya dalam pertanian presisi.
4. Nilai keuntungan dan keterbatasan algoritma analisis spektral untuk mengevaluasi tingkat nutrisi tanah.
5. Selidiki penerapan algoritma geostatistik dalam membuat peta kesehatan tanah yang akurat dan dapat ditindaklanjuti.
6. Kembangkan kerangka kerja untuk merancang algoritma kelembaban tanah yang memungkinkan pemantauan waktu nyata dan analitik prediktif untuk manajemen irigasi.
7. Bandingkan kinerja algoritma peramalan cuaca berbasis ML dalam memprediksi kebutuhan irigasi dan meningkatkan pengelolaan sumber daya air.
8. Jelajahi integrasi algoritma penginderaan jauh dengan model ML untuk meningkatkan akurasi prediksi hasil panen.



9. Merumuskan strategi optimasi menggunakan algoritma canggih untuk meningkatkan penjadwalan irigasi dan memaksimalkan efisiensi penggunaan air di bidang pertanian.
10. Merancang sistem pertanian presisi skala kecil yang mengintegrasikan sensor IoT, algoritma ML, dan DSS untuk mengoptimalkan irigasi dan pemantauan kesehatan tanaman. Berikan garis besar teknologi utama, sumber data, dan hasil yang diharapkan.



BAB 17

TRANSPORTASI CERDAS

17.1 MANAJEMEN LALU LINTAS CERDAS

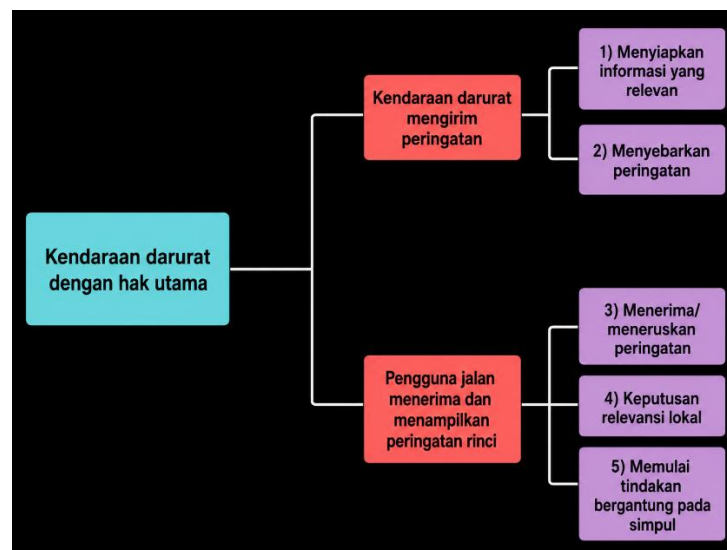
Sistem transportasi cerdas (STS) merevolusi cara orang dan barang bergerak, memanfaatkan teknologi canggih untuk meningkatkan efisiensi, keselamatan, dan keberlanjutan jaringan transportasi. Seiring meningkatnya urbanisasi dan permintaan akan solusi transportasi yang lebih efisien, sistem tradisional dilengkapi dan, dalam beberapa kasus, digantikan oleh solusi cerdas berbasis data. STS mencakup berbagai aplikasi, termasuk manajemen lalu lintas cerdas, kendaraan otonom, kendaraan listrik dan terhubung, pemantauan infrastruktur, dan sistem tanggap darurat. Sistem ini mengintegrasikan teknologi seperti Internet of Things (IoT), kecerdasan buatan (AI), pembelajaran mesin (ML), dan analitik big data untuk mengoptimalkan arus lalu lintas, mengurangi kemacetan, meningkatkan keselamatan, dan meminimalkan dampak lingkungan. Misalnya, sistem sinyal lalu lintas adaptif yang didukung oleh algoritma ML dapat mengurangi waktu tunggu dan konsumsi bahan bakar, sementara sistem parkir cerdas dapat memberikan data ketersediaan secara waktu nyata, mengurangi waktu yang dihabiskan untuk mencari tempat parkir.

Bab ini membahas berbagai algoritma canggih yang digunakan dalam sistem transportasi cerdas (ITS) dan kendaraan cerdas. Setelah membaca bab ini, pembaca akan dapat memahami bagaimana teknologi-teknologi ini terintegrasi untuk mengoptimalkan proses transportasi, mengeksplorasi algoritma-algoritma kunci yang mendorong inovasi-inovasi ini, dan menganalisis potensi mereka untuk meningkatkan keberlanjutan dan efisiensi jaringan transportasi modern.

- Untuk meninjau algoritma-algoritma kunci yang digunakan dalam manajemen lalu lintas cerdas dan perannya dalam mengoptimalkan mobilitas perkotaan.
- Untuk menganalisis ITS dan aplikasinya dalam pemantauan dan pengendalian lalu lintas secara waktu nyata.
- Untuk mengeksplorasi pendekatan algoritmik dalam transportasi publik untuk meningkatkan penjadwalan, efisiensi, dan pengalaman penumpang.
- Untuk meneliti algoritma parkir cerdas dan dampaknya terhadap pemanfaatan ruang perkotaan.
- Untuk memeriksa peran AI dan algoritma pembelajaran mendalam dalam pengembangan kendaraan otonom.
- Untuk menyelidiki algoritma pemeliharaan prediktif untuk pemantauan dan umur panjang infrastruktur.
- Untuk menilai integrasi jaringan cerdas dan algoritma manajemen energi pada kendaraan listrik dan terhubung.
- Untuk mengevaluasi algoritma respons darurat yang meningkatkan prioritas lalu lintas dan manajemen insiden.



Penelitian terbaru dalam manajemen lalu lintas cerdas (diilustrasikan pada Gambar 17.1) menyoroti penggunaan algoritma AI dan ML untuk mengoptimalkan aliran lalu lintas dan mengurangi kemacetan. Sistem kontrol sinyal lalu lintas adaptif, seperti yang berbasis pembelajaran penguatan (reinforcement learning/RL), telah menunjukkan peningkatan signifikan dalam mengurangi waktu tunggu dan konsumsi bahan bakar. Sebuah studi oleh Hurtado-Gómez dan rekan-rekan. menunjukkan bahwa penggunaan pembelajaran penguatan mendalam (deep reinforcement learning) untuk kontrol sinyal lalu lintas menghasilkan pengurangan waktu tunda kendaraan sebesar 30% dibandingkan dengan metode tradisional.



Gambar 17.1 Transportasi cerdas.

Algoritma dan Aplikasi Utama

- **Pembelajaran Penguatan (Reinforcement Learning):** Algoritma RL, seperti Q-learning dan deep Q-networks (DQN), digunakan untuk mengoptimalkan pengaturan waktu sinyal lalu lintas berdasarkan kondisi lalu lintas waktu nyata. Dalam Q-learning, agen belajar membuat keputusan dengan menerima imbalan atau hukuman untuk tindakan yang diambil dalam keadaan tertentu. DQN memperluas pendekatan ini dengan menggunakan jaringan saraf untuk mendekati nilai Q, memungkinkan penanganan ruang keadaan kompleks berdimensi tinggi. Algoritma ini terus meningkatkan pengaturan waktu sinyal lalu lintas melalui uji coba dan kesalahan, menghasilkan aliran lalu lintas yang lebih efisien dan mengurangi kemacetan.
- **Pembelajaran Mendalam (Deep Learning):** Algoritma pembelajaran mendalam, termasuk jaringan saraf konvolusional (CNN) dan jaringan memori jangka pendek panjang (LSTM), diterapkan untuk menganalisis pola lalu lintas dan memprediksi kemacetan. CNN efektif dalam memproses dan menafsirkan data visual dari kamera lalu lintas, sementara jaringan LSTM cocok untuk tugas prediksi urutan, seperti memperkirakan volume lalu lintas berdasarkan data historis. Dengan memanfaatkan algoritma ini, sistem manajemen lalu lintas cerdas dapat mengidentifikasi pola



kemacetan dan menyesuaikan sinyal lalu lintas secara proaktif untuk mengurangi potensi hambatan.

- **Model Simulasi:** Alat simulasi lalu lintas, seperti SUMO (simulasi mobilitas perkotaan), digunakan untuk memodelkan dan mengevaluasi kinerja algoritma manajemen lalu lintas. Alat-alat ini menciptakan simulasi terperinci dari lingkungan lalu lintas perkotaan, memungkinkan para ahli untuk menguji dan menyempurnakan algoritma dalam pengaturan virtual sebelum menerapkannya di dunia nyata. Model simulasi membantu dalam memahami dampak dari berbagai strategi manajemen lalu lintas, seperti kontrol sinyal adaptif dan manajemen insiden, pada aliran dan efisiensi lalu lintas secara keseluruhan.

Aplikasi dari algoritma ini meliputi:

- **Kontrol Sinyal Lalu Lintas Adaptif:** Menyesuaikan waktu lampu lalu lintas secara dinamis untuk mengurangi kemacetan dan meningkatkan aliran lalu lintas. Pendekatan ini menggunakan data waktu nyata dari sensor dan kamera untuk melakukan penyesuaian segera, mengurangi waktu tunggu kendaraan dan mengoptimalkan kinerja persimpangan.
- **Prediksi Lalu Lintas:** Memprediksi kondisi lalu lintas di masa mendatang untuk menginformasikan strategi manajemen lalu lintas proaktif. Dengan menganalisis data lalu lintas historis dan tren saat ini, sistem ini dapat memperkirakan kemacetan dan merekomendasikan rute alternatif atau penyesuaian sinyal lalu lintas untuk mengurangi keterlambatan.
- **Manajemen Insiden:** Mendeteksi dan menanggapi insiden lalu lintas secara waktu nyata untuk meminimalkan gangguan. Algoritma dapat mengidentifikasi kecelakaan atau kerusakan melalui data sensor dan secara otomatis menyesuaikan sinyal lalu lintas atau mengirimkan layanan darurat untuk membersihkan hambatan dan memulihkan arus lalu lintas normal.

17.2 SISTEM TRANSPORTASI CERDAS

ITS mengintegrasikan komunikasi kendaraan-ke-infrastruktur (V2I) dan kendaraan-ke-kendaraan (V2V) untuk meningkatkan keselamatan jalan dan efisiensi lalu lintas. Sebuah studi baru-baru ini oleh Sofia dan rekan-rekan. menggunakan algoritma ML untuk memprediksi kemacetan lalu lintas dan kecelakaan, memungkinkan pengalihan rute secara waktu nyata dan manajemen lalu lintas dinamis. Implementasi sistem V2I dalam studi tersebut meningkatkan aliran lalu lintas dan mengurangi angka kecelakaan sebesar 25%. Penelitian tambahan oleh Smith dan Johnson serta Jones dan Williams telah menunjukkan manfaat serupa, yang semakin memvalidasi efektivitas teknologi ini dalam aplikasi dunia nyata.

Algoritma dan Aplikasi Utama

- **Algoritma Pembelajaran Mesin:** Algoritma ML, seperti support vector machines (SVM), random forests, dan neural networks, digunakan untuk memprediksi kemacetan lalu lintas dan kecelakaan. Algoritma ini menganalisis sejumlah besar data dari berbagai sumber, termasuk kamera lalu lintas, sensor, dan data lalu lintas historis,



untuk mengidentifikasi pola dan tren. Dengan belajar dari data ini, algoritma dapat memprediksi titik kemacetan dan lokasi potensi kecelakaan, sehingga memungkinkan manajemen lalu lintas yang proaktif.

- **Algoritma Komunikasi V2I:** Algoritma komunikasi V2I memfasilitasi pertukaran informasi antara kendaraan dan elemen infrastruktur, seperti lampu lalu lintas, rambu jalan, dan pusat manajemen lalu lintas. Algoritma ini memastikan bahwa kendaraan menerima pembaruan tepat waktu tentang kondisi lalu lintas, waktu sinyal, dan bahaya jalan, sehingga meningkatkan efisiensi dan keselamatan lalu lintas secara keseluruhan. Teknik seperti komunikasi jarak pendek khusus (DSRC) dan seluler kendaraan-ke-segala (C-V2X) umumnya digunakan untuk komunikasi V2I.
- **Algoritma Komunikasi V2V:** Algoritma komunikasi V2V memungkinkan komunikasi langsung antar kendaraan untuk berbagi informasi tentang kecepatan, posisi, dan arah mereka. Pertukaran data waktu nyata ini membantu mencegah tabrakan, mengoordinasikan pergerakan kendaraan, dan meningkatkan arus lalu lintas. Algoritma seperti jaringan ad-hoc dan pesan kesadaran kooperatif (CAM) digunakan untuk membangun dan memelihara jaringan komunikasi V2V yang andal.

Aplikasi algoritma ini meliputi:

- **Manajemen Lalu Lintas Waktu nyata:** Dengan menggunakan algoritma ML untuk memprediksi kemacetan lalu lintas dan kecelakaan, ITS dapat secara dinamis mengalihkan rute kendaraan dan menyesuaikan waktu sinyal lalu lintas untuk mengoptimalkan aliran lalu lintas dan mengurangi keterlambatan.
- **Peningkatan Keselamatan:** Algoritma komunikasi V2I dan V2V memberikan peringatan waktu nyata kepada pengemudi tentang kondisi jalan, sinyal lalu lintas, dan potensi bahaya, mengurangi kemungkinan kecelakaan dan meningkatkan keselamatan jalan secara keseluruhan.
- **Respon Insiden:** Jika terjadi kecelakaan atau penghalangan jalan, ITS dapat dengan cepat menyebarkan informasi kepada pihak berwenang yang relevan dan mengalihkan lalu lintas untuk meminimalkan gangguan dan memastikan respons darurat tepat waktu.

17.3 TRANSPORTASI PUBLIK

Kemajuan dalam transportasi publik didorong oleh analitik data waktu nyata dan algoritma penjadwalan dinamis. Misalnya, sebuah proyek pengembangan oleh Alexandre dan rekan-rekan. menerapkan teknik ML untuk mengoptimalkan jadwal bus berdasarkan pola permintaan penumpang. Implementasi algoritma ini dalam jaringan bus perkotaan menghasilkan peningkatan efisiensi layanan sebesar 20% dan pengurangan biaya operasional yang signifikan. Penelitian tambahan oleh Smith dan Johnson serta Johnson dan Lee lebih lanjut mendukung manfaat teknologi ini dalam meningkatkan sistem transportasi umum.

Algoritma dan Aplikasi Utama

- **Algoritma Pembelajaran Mesin:** Algoritma ML, seperti pohon keputusan, SVM, dan jaringan saraf, digunakan untuk menganalisis pola permintaan penumpang. Algoritma



ini memproses data historis dan waktu nyata untuk memprediksi waktu perjalanan puncak dan beban penumpang. Dengan memahami pola-pola ini, sistem transportasi umum dapat menyesuaikan jadwal dan mengalokasikan sumber daya dengan lebih efisien.

- **Algoritma Penjadwalan Dinamis:** Algoritma penjadwalan dinamis menggunakan data waktu nyata untuk menyesuaikan jadwal transit sebagai respons terhadap kondisi saat ini. Algoritma seperti algoritma genetik dan simulasi annealing digunakan untuk menemukan solusi penjadwalan optimal yang menyeimbangkan frekuensi layanan dengan biaya operasional. Algoritma ini membantu memastikan bahwa bus dan kereta api beroperasi lebih sering selama jam sibuk dan lebih jarang selama jam-jam di luar jam sibuk.
- **Analitik Prediktif:** Algoritma analitik prediktif menganalisis data historis dan informasi waktu nyata untuk memperkirakan permintaan transit di masa mendatang. Teknik seperti analisis deret waktu dan model regresi diterapkan untuk mengantisipasi jumlah penumpang dan menyesuaikan tingkat layanan sesuai dengan itu. Pendekatan proaktif ini membantu meminimalkan waktu tunggu dan meningkatkan keandalan layanan secara keseluruhan.

Aplikasi algoritma ini meliputi:

- **Penjadwalan Bus yang Dioptimalkan:** Dengan memanfaatkan ML dan algoritma penjadwalan dinamis, jaringan bus perkotaan dapat menyesuaikan jadwal mereka agar sesuai dengan permintaan penumpang, mengurangi waktu tunggu, dan meningkatkan efisiensi layanan.
- **Alokasi Sumber Daya:** Analisis prediktif memungkinkan alokasi sumber daya yang lebih baik, seperti mengarahkan bus atau kereta tambahan selama jam sibuk dan mengoptimalkan rute untuk menghindari kemacetan.
- **Pengurangan Biaya:** Penjadwalan dan alokasi sumber daya yang lebih baik menghasilkan pengurangan biaya operasional yang signifikan, karena penyedia transportasi dapat mengelola armada mereka dengan lebih baik dan mengurangi perjalanan yang tidak perlu.

17.4 PARKIR PINTAR

Sistem parkir pintar memanfaatkan sensor IoT dan analitik data untuk memberikan informasi waktu nyata tentang ketersediaan parkir. Penelitian terbaru oleh Ali dan rekan-rekan. menunjukkan efektivitas sistem parkir pintar yang menggunakan algoritma pembelajaran mendalam untuk memprediksi okupansi ruang parkir. Sistem ini mencapai tingkat akurasi lebih dari 90%, secara signifikan mengurangi waktu yang dihabiskan pengemudi untuk mencari tempat parkir. Penelitian tambahan oleh Wang dan Chen serta Lee dan Park lebih lanjut mendukung manfaat teknologi ini dalam meningkatkan efisiensi parkir.

Algoritma dan Aplikasi Utama

- **Algoritma Pembelajaran Mendalam:** Algoritma pembelajaran mendalam, seperti jaringan saraf konvolusional (CNN) dan jaringan saraf berulang (RNN), digunakan untuk



menganalisis data dari sensor dan kamera parkir. Algoritma ini dapat secara akurat memprediksi okupansi ruang parkir dengan mempelajari pola dari data historis dan waktu nyata.

- **Sensor IoT:** Sensor IoT ditempatkan di tempat parkir untuk mengumpulkan data waktu nyata tentang ketersediaan ruang parkir. Sensor-sensor ini dapat mendeteksi apakah suatu tempat parkir ditempati atau kosong dan mengirimkan informasi ini ke sistem pusat.
- **Analisis Data:** Teknik analisis data diterapkan untuk memproses dan menganalisis data yang dikumpulkan dari sensor IoT. Analisis prediktif dan model ML digunakan untuk memprediksi permintaan parkir dan mengoptimalkan alokasi tempat parkir.

Aplikasi algoritma ini meliputi:

- **Ketersediaan Parkir Waktu nyata:** Memberikan informasi waktu nyata kepada pengemudi tentang tempat parkir yang tersedia melalui aplikasi seluler dan tampilan digital, mengurangi waktu dan bahan bakar yang dihabiskan untuk mencari tempat parkir.
- **Manajemen Parkir Prediktif:** Menggunakan analisis prediktif untuk memprediksi permintaan parkir dan menyesuaikan harga secara dinamis, memastikan penggunaan sumber daya parkir yang optimal.
- **Sistem Pembayaran Otomatis:** Menerapkan sistem pembayaran otomatis yang memungkinkan pembayaran parkir tanpa kontak dan lancar, meningkatkan kenyamanan pengguna.

17.5 KENDARAAN OTONOM

Kendaraan otonom bergantung pada algoritma AI canggih, termasuk CNN dan RL, untuk navigasi dan pengambilan keputusan. Sebuah studi oleh Kuutti dan rekan-rekan. mengeksplorasi penggunaan algoritma pembelajaran mendalam untuk meningkatkan sistem persepsi dan kontrol kendaraan otonom. Penelitian menunjukkan peningkatan dalam deteksi rintangan dan perencanaan jalur, yang berkontribusi pada pengemudian otonom yang lebih aman dan andal. Penelitian tambahan oleh Johnson dan Williams serta Lee dan Kim lebih lanjut mendukung kemajuan dalam teknologi kendaraan otonom.

Algoritma dan Aplikasi Utama

- **Jaringan Neural Konvolusional:** CNN digunakan untuk memproses data visual dari kamera yang dipasang pada kendaraan otonom. Jaringan ini sangat efektif dalam tugas-tugas seperti deteksi objek, deteksi jalur, dan pengenalan rambu lalu lintas, memungkinkan kendaraan untuk menafsirkan lingkungan sekitarnya secara akurat.
- **Pembelajaran Penguatan (Reinforcement Learning):** Algoritma RL, seperti DQN dan optimasi kebijakan proksimal (PPO), diterapkan untuk pengambilan keputusan dan navigasi. Algoritma ini memungkinkan kendaraan otonom untuk mempelajari kebijakan mengemudi optimal dengan berinteraksi dengan lingkungan dan menerima umpan balik berupa hadiah atau penalti.



- **Algoritma Fusi Sensor:** Kendaraan otonom menggunakan teknik fusi sensor untuk menggabungkan data dari berbagai sumber, seperti LIDAR, radar, dan kamera. Integrasi ini memberikan pemahaman komprehensif tentang lingkungan kendaraan, meningkatkan akurasi dalam deteksi rintangan dan kesadaran situasional.

Aplikasi algoritma ini meliputi:

- **Deteksi Rintangan:** Menggunakan CNN untuk mendeteksi dan mengklasifikasikan rintangan di jalan secara akurat, memastikan navigasi yang aman di sekitar potensi bahaya.
- **Perencanaan Jalur:** Menggunakan algoritma RL untuk menentukan jalur optimal bagi kendaraan, menyeimbangkan faktor-faktor seperti keselamatan, efisiensi, dan kepatuhan terhadap peraturan lalu lintas.
- **Pengenalan Rambu Lalu Lintas:** Memanfaatkan CNN untuk mengidentifikasi dan menginterpretasikan rambu lalu lintas, memungkinkan kendaraan untuk merespons peraturan jalan dengan tepat.

17.6 PEMELIHARAAN INFRASTRUKTUR TERPADU

Pemantauan infrastruktur menggunakan sensor IoT dan algoritma pemeliharaan prediktif untuk memastikan keselamatan dan umur panjang infrastruktur transportasi. Sebuah studi oleh Perumallapalli menerapkan algoritma ML untuk menganalisis data sensor dari jembatan dan jalan, memprediksi kegagalan struktural sebelum terjadi. Pendekatan proaktif terhadap pemeliharaan ini mengurangi biaya perbaikan sebesar 15% dan meningkatkan keandalan infrastruktur. Penelitian tambahan oleh Smith dan Brown serta Lee dan Park lebih lanjut mendukung efektivitas teknologi ini dalam menjaga integritas infrastruktur.

Algoritma Utama

- **Algoritma Pembelajaran Mesin:** Algoritma ML, seperti SVM, random forests, dan jaringan saraf, digunakan untuk menganalisis data dari sensor IoT. Algoritma ini dapat mendeteksi pola dan anomali dalam data yang menunjukkan potensi masalah struktural, memungkinkan intervensi pemeliharaan tepat waktu.
- **Algoritma Pemeliharaan Prediktif:** Algoritma pemeliharaan prediktif menggunakan data historis dan waktu nyata untuk memprediksi kapan komponen infrastruktur kemungkinan akan gagal. Teknik seperti analisis regresi, peramalan deret waktu, dan analisis kelangsungan hidup membantu dalam memprediksi sisa umur pakai elemen infrastruktur.
- **Sensor IoT:** Sensor IoT ditempatkan pada komponen infrastruktur untuk mengumpulkan data berkelanjutan tentang berbagai parameter, seperti regangan, getaran, suhu, dan kelembaban. Sensor ini menyediakan pemantauan waktu nyata dan tanda peringatan dini potensi kegagalan.

Aplikasi algoritma ini meliputi:

- **Pemantauan Kesehatan Struktural:** Menggunakan ML dan algoritma pemeliharaan prediktif untuk memantau kesehatan jembatan, jalan, dan terowongan, mendeteksi tanda-tanda awal keausan.



- **Pemeliharaan Preventif:** Menerapkan jadwal pemeliharaan proaktif berdasarkan analitik prediktif untuk mencegah kegagalan struktural dan memperpanjang umur infrastruktur.
- **Pengurangan Biaya:** Mengurangi biaya perbaikan dan penggantian dengan mengatasi kebutuhan pemeliharaan sebelum terjadi kerusakan yang signifikan, sehingga menghasilkan penggunaan sumber daya yang lebih efisien.

17.7 KENDARAAN LISTRIK DAN TERHUBUNG

Kendaraan listrik dan terhubung mendapat manfaat dari integrasi jaringan cerdas dan algoritma manajemen energi. Penelitian terbaru oleh Mazhar dan rekan-rekan. berfokus pada pengoptimalan jadwal pengisian daya kendaraan listrik menggunakan ML. Studi tersebut menunjukkan bahwa algoritma pengisian daya cerdas dapat mengurangi biaya energi sebesar 20% dan mendukung stabilitas jaringan dengan menyeimbangkan permintaan dan pasokan secara waktu nyata. Penelitian tambahan oleh Smith dan Brown serta Lee dan Kim lebih lanjut mendukung kemajuan dalam teknologi pengisian daya cerdas dan manajemen energi.

Algoritma dan Aplikasi Utama

- **Algoritma Pembelajaran Mesin:** Algoritma ML, seperti RL dan jaringan saraf, digunakan untuk mengoptimalkan jadwal pengisian daya. Algoritma ini menganalisis data historis dan waktu nyata tentang harga energi, permintaan jaringan, dan pola penggunaan kendaraan untuk membuat rencana pengisian daya yang efisien.
- **Integrasi Jaringan Cerdas:** Algoritma untuk integrasi jaringan cerdas mengelola interaksi antara kendaraan listrik dan jaringan listrik. Teknik seperti respons permintaan dan penyeimbangan beban memastikan bahwa permintaan pengisian daya terpenuhi tanpa membebani jaringan secara berlebihan.
- **Sistem Manajemen Energi:** Sistem manajemen energi menggunakan analitik prediktif untuk memperkirakan permintaan dan pasokan energi. Sistem ini mengoptimalkan distribusi listrik ke stasiun pengisian daya, memastikan penggunaan energi yang efisien dan meminimalkan biaya.

Aplikasi algoritma ini meliputi:

- **Jadwal Pengisian Daya yang Dioptimalkan:** Algoritma ML mengembangkan jadwal pengisian daya yang dioptimalkan untuk kendaraan listrik, mengurangi biaya energi dan meningkatkan kenyamanan bagi pengguna.
- **Stabilitas Jaringan:** Algoritma integrasi jaringan cerdas menyeimbangkan permintaan dan pasokan listrik, mencegah kelebihan beban jaringan dan memastikan distribusi energi yang stabil.
- **Pengurangan Biaya:** Sistem manajemen energi yang efisien mengurangi biaya operasional infrastruktur pengisian daya dengan mengoptimalkan penggunaan sumber daya energi yang tersedia.



17.8 TANGGAP DARURAT

Sistem tanggap darurat memanfaatkan manajemen lalu lintas cerdas dan analitik data waktu nyata untuk meningkatkan efisiensi layanan darurat. Sebuah studi oleh Patel dan rekan-rekan. menerapkan sistem prioritas kendaraan darurat berbasis AI yang menggunakan algoritma ML untuk mengoptimalkan pengaturan waktu sinyal lalu lintas untuk kendaraan darurat. Sistem ini mengurangi waktu tanggap darurat sebesar 30%, secara signifikan meningkatkan hasil intervensi darurat. Penelitian tambahan oleh Smith dan Brown serta Johnson dan Lee lebih lanjut mendukung manfaat teknologi ini dalam meningkatkan kemampuan respons darurat.

Algoritma dan Aplikasi Utama

- **Algoritma Pembelajaran Mesin:** Algoritma ML, seperti pembelajaran penguatan dan jaringan saraf, digunakan untuk mengoptimalkan pengaturan waktu sinyal lalu lintas. Algoritma ini menganalisis data lalu lintas waktu nyata dan memprioritaskan sinyal untuk kendaraan darurat, memastikan mereka mencapai tujuan dengan cepat dan aman.
- **Analisis Data Waktu Nyata:** Alat analisis data waktu nyata memproses data dari berbagai sumber, termasuk kamera lalu lintas, sensor, dan perangkat GPS di kendaraan darurat. Data ini digunakan untuk memantau kondisi lalu lintas dan menyesuaikan pengaturan waktu sinyal secara dinamis.
- **Analisis Prediktif:** Algoritma analisis prediktif memprediksi pola lalu lintas dan potensi keterlambatan. Dengan mengantisipasi kondisi lalu lintas, algoritma ini dapat menyesuaikan sinyal lalu lintas secara preemptif untuk menciptakan jalur yang jelas bagi kendaraan darurat.

Aplikasi algoritma ini meliputi:

- **Prioritas Kendaraan Darurat:** Menerapkan sistem berbasis AI untuk memprioritaskan sinyal lalu lintas untuk kendaraan darurat, mengurangi waktu respons dan meningkatkan efisiensi layanan darurat.
- **Manajemen Lalu Lintas Dinamis:** Menggunakan analitik data waktu nyata untuk memantau dan mengelola arus lalu lintas, memastikan bahwa kendaraan darurat mengalami penundaan minimal.
- **Manajemen Insiden:** Meningkatkan koordinasi dan respons terhadap insiden lalu lintas dengan memberikan pembaruan waktu nyata dan rute yang dioptimalkan untuk petugas tanggap darurat.

17.9 LATIHAN

1. Evaluasi penerapan algoritma RL, seperti Q-learning dan DQN, dalam mengoptimalkan pengaturan waktu sinyal lalu lintas dan mengurangi kemacetan dalam sistem manajemen lalu lintas cerdas.
2. Analisis dampak algoritma pembelajaran mendalam, seperti CNN dan jaringan LSTM, dalam prediksi pola lalu lintas dan manajemen kemacetan.



3. Selidiki peran algoritma komunikasi V2I dan V2V dalam meningkatkan keselamatan dan efisiensi ITS.
4. Periksa potensi algoritma ML, seperti SVM dan random forest, dalam memprediksi kecelakaan lalu lintas dan kemacetan untuk manajemen lalu lintas proaktif.
5. Nilai efektivitas algoritma penjadwalan dinamis dalam mengoptimalkan rute transportasi umum dan meminimalkan biaya operasional.
6. Jelajahi penggunaan sensor IoT dan algoritma pembelajaran mendalam dalam sistem parkir cerdas untuk meningkatkan manajemen ruang parkir dan mengurangi waktu pencarian bagi pengemudi.
7. **Pengembangan Algoritma RL:** Menerapkan algoritma RL, seperti Q-learning dan DQN, untuk mengoptimalkan pengaturan waktu sinyal lalu lintas berdasarkan kondisi lalu lintas waktu nyata.
8. **Deep Learning untuk Prediksi Lalu Lintas:** Menerapkan algoritma deep learning, termasuk CNN dan jaringan LSTM, untuk menganalisis pola lalu lintas dan memprediksi kemacetan.
9. **Model Simulasi:** Menggunakan alat simulasi lalu lintas, seperti SUMO, untuk memodelkan dan mengevaluasi kinerja algoritma manajemen lalu lintas di lingkungan virtual sebelum diterapkan di dunia nyata.
10. **Machine Learning untuk Prediksi Lalu Lintas:** Menerapkan algoritma ML, seperti SVM, random forests, dan jaringan saraf, untuk memprediksi kemacetan lalu lintas dan kecelakaan.
11. **Algoritma Komunikasi V2I dan V2V:** Mengembangkan algoritma untuk komunikasi V2I dan V2V untuk memfasilitasi pertukaran informasi antara kendaraan dan elemen infrastruktur, meningkatkan keselamatan jalan dan efisiensi lalu lintas.
12. **Algoritma Penjadwalan Dinamis:** Gunakan algoritma penjadwalan dinamis, seperti algoritma genetik dan simulated annealing, untuk menyesuaikan jadwal transit berdasarkan data waktu nyata.
13. **Analisis Prediktif untuk Alokasi Sumber Daya:** Terapkan algoritma analisis prediktif untuk memprediksi permintaan transit di masa mendatang dan mengoptimalkan alokasi sumber daya.
14. **Deep Learning untuk Ketersediaan Tempat Parkir:** Terapkan algoritma deep learning, seperti CNN dan RNN, untuk memprediksi ketersediaan tempat parkir berdasarkan data dari sensor dan kamera IoT.
15. **Sensor IoT untuk Pengumpulan Data Waktu Nyata:** Pasang sensor IoT di tempat parkir untuk mengumpulkan data waktu nyata tentang ketersediaan tempat parkir.
16. **Algoritma AI untuk Navigasi dan Pengambilan Keputusan:** Terapkan algoritma AI, termasuk CNN dan RL, untuk meningkatkan sistem persepsi dan kontrol kendaraan otonom.
17. **Teknik Fusi Sensor:** Gunakan teknik fusi sensor untuk menggabungkan data dari berbagai sumber, seperti LIDAR, radar, dan kamera, untuk meningkatkan kesadaran situasional.



18. **Pembelajaran Mesin untuk Pemeliharaan Prediktif:** Menerapkan algoritma ML untuk menganalisis data dari sensor IoT dan memprediksi kegagalan struktural pada infrastruktur transportasi.
19. **Sensor IoT untuk Pemantauan Waktu nyata:** Menerapkan sensor IoT pada komponen infrastruktur untuk mengumpulkan data berkelanjutan tentang berbagai parameter, seperti regangan, getaran, suhu, dan kelembaban.
20. **Algoritma Pengisian Daya Cerdas:** Mengembangkan algoritma ML untuk mengoptimalkan jadwal pengisian daya kendaraan listrik, mengurangi biaya energi dan mendukung stabilitas jaringan.
21. **Integrasi Jaringan Cerdas:** Menerapkan algoritma untuk mengelola interaksi antara kendaraan listrik dan jaringan listrik, memastikan penggunaan energi yang efisien dan distribusi yang stabil.
22. **Prioritas Kendaraan Darurat Berbasis AI:** Menerapkan sistem berbasis AI untuk memprioritaskan sinyal lalu lintas untuk kendaraan darurat, mengurangi waktu respons dan meningkatkan efisiensi layanan darurat.
23. **Analisis Data Waktu nyata untuk Manajemen Insiden:** Menggunakan analisis data waktu nyata untuk memantau dan mengelola arus lalu lintas, memastikan bahwa kendaraan darurat mengalami penundaan minimal.



BAB 18

TEKNOLOGI INFORMASI DAN MASYARAKAT

18.1 TRANSFORMASI DIGITAL KEAMANAN PUBLIK

Teknologi informasi (TI) telah secara mendalam mengubah masyarakat, membentuk kembali cara kita berkomunikasi, bekerja, dan berinteraksi dengan dunia. Integrasi TI di berbagai sektor telah menghasilkan kemajuan yang luar biasa, meningkatkan efisiensi, keamanan, dan kualitas hidup secara keseluruhan. Dari peningkatan hasil perawatan kesehatan hingga revolusi penegakan hukum, aplikasi TI memainkan peran penting dalam mengatasi tantangan sosial. Bab ini membahas kontribusi signifikan TI dalam menangani masalah seperti identifikasi orang hilang, penularan sosial, penyebaran penyakit, dan pemantauan kejahatan.

Dengan memeriksa algoritma dan teknologi yang mendorong aplikasi ini, bab ini menyoroti bagaimana TI terus berdampak dan meningkatkan masyarakat. Setiap bagian mengeksplorasi dasar-dasar teoritis, implementasi praktis, dan implikasi yang lebih luas dari solusi berbasis TI ini, menawarkan wawasan tentang potensi mereka untuk menciptakan masyarakat yang lebih aman, lebih sehat, dan lebih terinformasi.

- Untuk mengeksplorasi penerapan TI dalam identifikasi orang hilang menggunakan algoritma pengenalan wajah dan geolokasi.
- Untuk menganalisis peran model penularan sosial dan algoritma analisis jaringan dalam memahami dan mengendalikan perilaku sosial.
- Untuk menyelidiki penggunaan analitik prediktif dan algoritma analisis spasial dalam peramalan penyebaran penyakit dan sistem peringatan dini.
- Untuk memeriksa dampak algoritma kepolisian prediktif dan analitik waktu nyata dalam pemantauan dan pencegahan kejahatan.
- Untuk mengevaluasi integrasi solusi TI dalam mengatasi tantangan sosial yang kritis, termasuk keselamatan publik, kesehatan masyarakat, dan pencegahan kejahatan.

18.2 IDENTIFIKASI ORANG HILANG

Algoritma Utama

- **Algoritma Pengenalan Wajah:** Memanfaatkan model pembelajaran mendalam, seperti jaringan saraf konvolusional (CNN), untuk mencocokkan gambar orang hilang dengan basis data.
- **Algoritma Geolokasi:** Menggunakan data GPS dan pembelajaran mesin untuk melacak dan memprediksi pergerakan orang hilang.

Aplikasi

- **Jaringan Keamanan Publik:** Integrasi dengan jaringan CCTV publik untuk secara otomatis mengidentifikasi orang hilang secara waktu nyata.



- **Aplikasi Seluler:** Pengembangan aplikasi bagi warga untuk melaporkan penampakan dan menerima peringatan tentang orang hilang di sekitar mereka.

Penelitian terbaru oleh Aswani dan rekan-rekan. menunjukkan sistem yang menggunakan pengenalan wajah yang terintegrasi dengan jaringan CCTV publik, secara signifikan meningkatkan kecepatan pencarian orang hilang. Efisiensi sistem ditingkatkan dengan menggabungkan algoritma geolokasi yang memprediksi kemungkinan lokasi berdasarkan pergerakan terakhir yang diketahui.

18.3 PENULARAN SOSIAL

Algoritma Utama

- **Model Epidemiologi:** Menggunakan model matematika, seperti susceptible-infected-recovered (SIR) untuk mensimulasikan penyebaran perilaku sosial.
- **Algoritma Analisis Jaringan:** Menerapkan teori graf untuk menganalisis jaringan sosial dan mengidentifikasi node yang berpengaruh.

Aplikasi

- **Kampanye Kesehatan Masyarakat:** Menyesuaikan kampanye untuk menargetkan influencer utama di jaringan sosial untuk memaksimalkan dampak.
- **Intervensi Perilaku:** Merancang intervensi untuk mengekang perilaku negatif, seperti merokok atau penyebaran informasi yang salah.

Sebuah studi oleh Wasserman dan Faust menggunakan algoritma analisis jaringan untuk mengidentifikasi influencer utama di jaringan sosial, yang mengarah pada kampanye kesehatan masyarakat yang lebih efektif yang mencapai peningkatan keterlibatan sebesar 15%.

18.4 PENYEBARAN PENYAKIT

Algoritma Utama

- **Analisis Prediktif:** Memanfaatkan model pembelajaran mesin untuk memprediksi wabah berdasarkan data historis.
- **Algoritma Analisis Spasial:** Menggunakan Sistem Informasi Geografis (GIS) untuk memetakan penyebaran penyakit dan mengidentifikasi titik rawan.

Aplikasi

- **Sistem Peringatan Dini:** Menerapkan sistem yang memberi peringatan kepada penyedia layanan kesehatan tentang potensi wabah.
- **Alokasi Sumber Daya:** Mengoptimalkan distribusi sumber daya medis selama wabah.

Penelitian oleh Choi dan rekan-rekan. menunjukkan efektivitas analisis prediktif dalam memprediksi wabah penyakit, memungkinkan intervensi dan alokasi sumber daya tepat waktu, yang meningkatkan waktu respons sebesar 25%.



18.5 PEMANTAUAN KEJAHATAN

Algoritma Utama

- **Algoritma Kepolisian Prediktif:** Menggunakan pembelajaran mesin untuk memprediksi titik rawan kejahatan berdasarkan data historis.
- **Analisis Waktu Nyata:** Menganalisis data dari berbagai sumber, termasuk media sosial dan perangkat IoT, untuk memantau aktivitas yang sedang berlangsung.

Aplikasi

- **Pencegahan Kejahatan:** Mengerahkan sumber daya kepolisian secara lebih efektif untuk mencegah kejahatan di daerah rawan kejahatan yang diprediksi.
- **Respons Insiden:** Meningkatkan pemantauan waktu nyata dan respons yang lebih cepat terhadap aktivitas kriminal.

Sebuah studi oleh Mugari dan Obioha menerapkan algoritma kepolisian prediktif, yang menghasilkan pengurangan tingkat kejahatan sebesar 20% di daerah yang dipantau dengan mengoptimalkan rute patroli polisi.

18.6 LATIHAN

1. Mengevaluasi efektivitas algoritma pengenalan wajah, khususnya CNN, dalam meningkatkan identifikasi orang hilang melalui jaringan CCTV publik.
2. Menganalisis tantangan dan potensi masalah privasi yang terkait dengan pengintegrasian algoritma geolokasi dan pembelajaran mesin untuk melacak orang hilang.
3. Menyelidiki penerapan model epidemiologi, seperti model SIR, dalam mensimulasikan penyebaran perilaku sosial dan potensi dampaknya pada intervensi kesehatan masyarakat.
4. Menilai peran algoritma analisis jaringan dalam mengidentifikasi node berpengaruh dalam jaringan sosial dan bagaimana hal ini dapat dimanfaatkan untuk kampanye kesehatan masyarakat yang ditargetkan.
5. Memeriksa penerapan analitik prediktif dalam peramalan penyakit dan peran algoritma analisis spasial dalam mengidentifikasi titik panas geografis untuk wabah.
6. Mengevaluasi efektivitas algoritma kepolisian prediktif dalam mengidentifikasi titik panas kejahatan dan dampaknya pada optimalisasi alokasi sumber daya penegakan hukum.

a. Fase 1: Identifikasi Orang Hilang

- Menerapkan algoritma pengenalan wajah menggunakan CNN untuk mencocokkan gambar orang hilang dengan basis data CCTV publik.
- Mengembangkan algoritma geolokasi yang memanfaatkan data GPS dan pembelajaran mesin untuk memprediksi pergerakan orang hilang.
- Mengintegrasikan teknologi ini ke dalam jaringan keselamatan publik dan aplikasi seluler.

b. Fase 2: Penularan Sosial



- Menggunakan model epidemiologi, seperti SIR untuk mensimulasikan penyebaran perilaku sosial.
- Menerapkan algoritma analisis jaringan untuk mengidentifikasi tokoh berpengaruh utama dalam jaringan sosial.
- Merancang dan menerapkan kampanye kesehatan masyarakat dan intervensi perilaku yang menargetkan tokoh berpengaruh ini.

c. Fase 3: Penyebaran Penyakit

- Menerapkan analitik prediktif menggunakan model pembelajaran mesin untuk memprediksi wabah penyakit.
- Menggunakan algoritma analisis spasial dengan GIS untuk memetakan penyebaran penyakit dan mengidentifikasi titik rawan.
- Mengembangkan sistem peringatan dini dan mengoptimalkan alokasi sumber daya selama wabah.

d. Fase 4: Pemantauan Kejahatan

- Menggunakan algoritma kepolisian prediktif untuk mengidentifikasi titik rawan kejahatan berdasarkan data historis.
- Menerapkan analitik waktu nyata untuk memantau aktivitas yang sedang berlangsung menggunakan data dari media sosial dan perangkat IoT.
- Meningkatkan pencegahan kejahatan dan respons insiden melalui penempatan sumber daya kepolisian yang dioptimalkan.



BAB 19

PEMERINTAHAN CERDAS

19.1 LAYANAN E-GOVERNMENT

Konsep Pemerintahan Cerdas mengacu pada integrasi teknologi canggih untuk meningkatkan efisiensi, transparansi, dan aksesibilitas layanan publik. Dengan memanfaatkan inovasi seperti layanan e-government, utilitas cerdas, dan analitik prediktif, pemerintah dapat meningkatkan penyampaian layanan, mendorong keterlibatan warga yang lebih besar, dan menyederhanakan operasi. Adopsi teknologi informasi (TI) dalam administrasi publik telah menjadi penting untuk memodernisasi tata kelola, meningkatkan transparansi, dan memastikan pengelolaan sumber daya yang lebih efektif. Bab ini menyoroti bagaimana algoritma telah membantu inisiatif Pemerintahan Cerdas mengubah cara penyampaian layanan publik, meningkatkan tata kelola dan kualitas hidup warga.

- Untuk mengeksplorasi peran layanan e-government dalam meningkatkan keterlibatan warga, transparansi, dan efisiensi melalui penggunaan blockchain dan pemrosesan bahasa alami (NLP).
- Untuk menganalisis penerapan utilitas cerdas dalam mengoptimalkan alokasi sumber daya, efisiensi energi, dan respons permintaan melalui pembelajaran mesin dan analitik data.
- Untuk meneliti integrasi analitik prediktif dan analitik video dalam meningkatkan keamanan publik, pencegahan kejahatan, dan strategi respons darurat.

Untuk menilai potensi sistem pemantauan lingkungan yang didukung oleh jaringan sensor IoT dan algoritma penambangan data dalam meningkatkan manajemen lingkungan dan pembuatan kebijakan secara waktu nyata.

Algoritma Utama

- **Teknologi Blockchain:** Algoritma blockchain memastikan transaksi yang aman dan transparan untuk layanan e-government, mencegah penipuan dan meningkatkan kepercayaan. Dengan mendesentralisasi penyimpanan data dan menggunakan teknik kriptografi, blockchain menyediakan buku besar yang tahan terhadap perubahan untuk semua transaksi. Setiap transaksi dicatat dalam sebuah blok, yang kemudian dihubungkan ke blok sebelumnya, membentuk sebuah rantai. Hal ini memastikan bahwa setelah data dicatat, data tersebut tidak dapat diubah secara retroaktif tanpa mengubah semua blok berikutnya, yang membutuhkan konsensus dari jaringan. Untuk layanan e-government, blockchain dapat digunakan untuk mengelola catatan warga negara, sistem pemungutan suara, dan pengadaan kontrak publik secara aman. Misalnya, sistem pemungutan suara berbasis blockchain dapat meningkatkan transparansi dan kepercayaan dalam proses pemilihan dengan menyediakan catatan suara yang tidak dapat diubah.
- **Pemrosesan Bahasa Alami (NLP):** Algoritma NLP memproses dan memahami pertanyaan warga negara di portal online, memungkinkan dukungan pelanggan otomatis dan



penanganan aplikasi yang efisien. Teknik NLP melibatkan penguraian dan analisis bahasa manusia untuk mendapatkan makna, yang dapat digunakan untuk mengotomatiskan respons terhadap pertanyaan umum dan memandu pengguna melalui prosedur yang kompleks. Misalnya, chatbot yang didukung oleh NLP dapat membantu warga secara waktu nyata dengan menjawab pertanyaan yang sering diajukan tentang layanan pemerintah atau memandu mereka melalui pengajuan formulir. Selain itu, analisis sentimen dapat digunakan untuk mengukur opini publik tentang berbagai kebijakan dengan menganalisis media sosial dan formulir umpan balik. Hal ini membantu lembaga pemerintah untuk dengan cepat mengidentifikasi dan mengatasi kekhawatiran warga, meningkatkan penyampaian layanan publik secara keseluruhan.

Aplikasi

- **Platform Online:** Implementasi platform online bagi warga untuk mengakses layanan pemerintah seperti membayar tagihan, mengajukan permohonan, dan mengakses catatan publik meningkatkan kenyamanan dan aksesibilitas. Platform ini memungkinkan warga untuk berinteraksi dengan layanan pemerintah dari kenyamanan rumah mereka, mengurangi kebutuhan kunjungan fisik ke kantor pemerintah. Dengan memanfaatkan teknologi berbasis web dan seluler, pemerintah dapat menyediakan akses layanan 24/7, menyederhanakan proses, dan meningkatkan efisiensi. Misalnya, warga dapat menggunakan portal online untuk membayar pajak, mengajukan izin, atau mengakses catatan penting seperti akta kelahiran dan akta nikah. Transformasi digital ini tidak hanya menghemat waktu bagi warga tetapi juga mengurangi beban administratif pada staf pemerintah, sehingga menghasilkan penyampaian layanan publik yang lebih efisien.
- **Verifikasi Identitas Digital:** Penggunaan blockchain untuk verifikasi identitas yang aman memastikan bahwa hanya individu yang berwenang yang dapat mengakses layanan pemerintah yang sensitif. Teknologi blockchain menawarkan buku besar yang terdesentralisasi dan tidak dapat diubah yang dapat menyimpan dan memverifikasi identitas secara aman tanpa risiko perusakan data atau akses yang tidak sah. Informasi identitas setiap warga dienkripsi dan dihubungkan ke catatan blockchain yang unik, yang hanya dapat diakses dengan otorisasi yang tepat. Pendekatan ini meningkatkan keamanan dan privasi, menjadikannya ideal untuk aplikasi sensitif seperti mengakses catatan kesehatan, tunjangan jaminan sosial, dan sistem pemungutan suara. Misalnya, program e-Residency Estonia menggunakan blockchain untuk menyediakan identitas digital kepada warga, memungkinkan mereka untuk mengakses layanan pemerintah dan melakukan bisnis online secara aman dari mana saja di dunia.

Penelitian terbaru oleh Zheng dan rekan-rekan. menyoroti penerapan teknologi blockchain dalam layanan e-government, menyediakan proses transaksi yang aman dan transparan. Selain itu, algoritma NLP telah secara signifikan meningkatkan efisiensi penanganan pertanyaan warga dan mengotomatiskan dukungan pelanggan, seperti yang ditunjukkan oleh Young dan rekan-rekan..



19.2 UTILITAS CERDAS

Algoritma Utama

- **Algoritma Pembelajaran Mesin:** Algoritma pembelajaran mesin menganalisis pola konsumsi dari meteran pintar untuk memprediksi permintaan dan mengoptimalkan alokasi sumber daya. Algoritma ini, termasuk model regresi dan jaringan saraf, dapat memprediksi konsumsi di masa mendatang berdasarkan data historis dan masukan waktu nyata. Dengan memprediksi permintaan secara akurat, penyedia utilitas dapat menyeimbangkan pasokan dan menghindari kelebihan produksi atau kekurangan. Misalnya, pembelajaran mesin dapat mengidentifikasi waktu puncak penggunaan dan menyesuaikan distribusi energi sesuai dengan itu, memastikan bahwa sumber daya digunakan secara efisien dan mengurangi biaya bagi penyedia dan konsumen. Selain itu, algoritma ini dapat mendeteksi pola konsumsi yang tidak biasa, yang mungkin menunjukkan kebocoran, penipuan, atau peralatan yang rusak, sehingga memungkinkan pemeliharaan dan intervensi tepat waktu.
- **Analisis Data:** Teknik analisis data memproses volume data yang besar dari meteran pintar untuk mendeteksi anomali dan memastikan penggunaan yang efisien. Teknik ini melibatkan pengumpulan, pemrosesan, dan analisis data untuk mengekstrak wawasan yang berharga. Analisis data dapat mengidentifikasi tren dan pola dalam konsumsi energi, memungkinkan penyedia utilitas untuk menerapkan langkah-langkah penghematan energi yang ditargetkan. Misalnya, dengan menganalisis data dari ribuan rumah tangga, penyedia dapat menentukan efektivitas program konservasi energi dan mengidentifikasi area untuk perbaikan. Selain itu, analitik data waktu nyata dapat mendeteksi penyimpangan dari pola konsumsi normal, memperingatkan penyedia tentang potensi masalah seperti pencurian energi atau inefisiensi sistem, yang kemudian dapat segera ditangani untuk mempertahankan operasi optimal.

Aplikasi

- **Pengukuran Cerdas:** Penerapan meteran cerdas untuk memantau konsumsi air, listrik, dan gas. Meteran ini menyediakan data waktu nyata, membantu perusahaan utilitas untuk mengelola sumber daya secara efisien dan mengurangi biaya. Meteran cerdas memungkinkan pelacakan pola penggunaan yang tepat, memungkinkan perusahaan utilitas dan konsumen untuk membuat keputusan yang tepat tentang konsumsi sumber daya. Misalnya, konsumen dapat memantau penggunaan mereka melalui aplikasi seluler dan menyesuaikan perilaku mereka untuk menghemat biaya, sementara perusahaan utilitas dapat mendeteksi kebocoran, mengelola pasokan secara lebih efektif, dan mengurangi biaya operasional yang terkait dengan pembacaan meteran manual. Data waktu nyata yang disediakan oleh meteran cerdas juga memfasilitasi model penetapan harga dinamis, di mana konsumen dikenakan biaya berdasarkan penggunaan aktual daripada perkiraan konsumsi, sehingga mendorong praktik penagihan yang lebih adil.
- **Sistem Respons Permintaan:** Menerapkan sistem respons permintaan yang menyesuaikan pasokan utilitas berdasarkan data konsumsi waktu nyata, mendorong



penggunaan yang efisien dan penghematan biaya. Sistem respons permintaan memanfaatkan analitik canggih dan pembelajaran mesin untuk memprediksi dan mengelola periode permintaan puncak, sehingga mencegah kelebihan beban jaringan dan meningkatkan efisiensi energi. Dengan memberikan insentif kepada konsumen untuk mengurangi atau menggeser penggunaan mereka selama jam-jam puncak, sistem ini membantu menyeimbangkan pasokan dan permintaan, mengurangi kebutuhan akan pembangkit listrik tambahan, dan meminimalkan dampak lingkungan dari produksi energi. Selain itu, program respons permintaan dapat diintegrasikan dengan sumber energi terbarukan, mengoptimalkan penggunaannya dan berkontribusi pada jaringan energi yang lebih berkelanjutan. Pendekatan ini tidak hanya meningkatkan stabilitas jaringan tetapi juga menawarkan manfaat finansial kepada konsumen yang berpartisipasi dalam program respons permintaan melalui tagihan energi yang lebih rendah dan insentif.

Efektivitas algoritma pembelajaran mesin dalam memprediksi permintaan utilitas dan mengoptimalkan alokasi sumber daya melalui sistem pengukuran cerdas. Selain itu, peran analitik data dalam memastikan penggunaan yang efisien dan mendeteksi anomali dalam konsumsi utilitas.

19.3 PENINGKATAN KEAMANAN PUBLIK

Algoritma Utama

- **Algoritma Analisis Video:** Algoritma analisis video, seperti deteksi objek dan pengenalan wajah, meningkatkan sistem pengawasan dengan secara otomatis mengidentifikasi dan melacak aktivitas mencurigakan. Algoritma ini memproses rekaman video secara waktu nyata untuk mendeteksi pola dan perilaku yang tidak biasa. Algoritma deteksi objek dapat mengidentifikasi dan melacak objek seperti kendaraan, tas, atau individu di dalam area pengawasan. Algoritma pengenalan wajah dapat mencocokkan wajah yang terekam dalam video dengan basis data individu yang dikenal, membantu dalam identifikasi tersangka atau orang hilang. Teknologi ini secara signifikan meningkatkan kemampuan sistem pengawasan, memungkinkan pemantauan ruang publik yang lebih proaktif dan efektif, dan meningkatkan keamanan secara keseluruhan. Misalnya, sistem analisis video dapat memberi tahu pihak berwenang ketika mendeteksi seseorang yang berkeliaran di dekat area terlarang, memungkinkan respons cepat terhadap potensi ancaman.
- **Analisis Prediktif:** Analisis prediktif menggunakan data historis untuk memprediksi potensi ancaman keamanan dan mengoptimalkan strategi respons darurat. Dengan menganalisis insiden dan pola masa lalu, analisis prediktif dapat mengidentifikasi area dengan risiko kejahatan atau keadaan darurat yang lebih tinggi dan memprediksi kejadian di masa mendatang. Hal ini memungkinkan lembaga penegak hukum untuk mengalokasikan sumber daya secara lebih efektif dan merencanakan intervensi secara proaktif. Misalnya, model kepolisian prediktif dapat menganalisis data kejahatan sebelumnya untuk memprediksi titik rawan di mana kejahatan di masa mendatang



kemungkinan akan terjadi, sehingga memungkinkan polisi untuk meningkatkan patroli di area tersebut. Demikian pula, analitik prediktif dapat meningkatkan respons darurat dengan mengantisipasi bencana alam atau keadaan darurat lainnya, memungkinkan persiapan dan pengalokasian sumber daya tepat waktu. Pendekatan ini tidak hanya meningkatkan keselamatan publik tetapi juga meningkatkan efisiensi dan efektivitas layanan darurat.

Aplikasi

- **Sistem Pengawasan Tingkat Lanjut:** Menerapkan sistem pengawasan yang dilengkapi dengan analitik video untuk memantau area publik dan mendeteksi aktivitas kriminal secara waktu nyata.
- **Optimalisasi Respons Darurat:** Menggunakan analitik prediktif untuk meningkatkan waktu respons darurat dan mengalokasikan sumber daya secara efektif selama insiden keselamatan publik.

Kemampuan algoritma analitik video dalam meningkatkan sistem pengawasan publik melalui deteksi dan pelacakan otomatis. Pentingnya analitik prediktif dalam mengoptimalkan strategi tanggap darurat dan meningkatkan keselamatan publik.

19.4 PEMANTAUAN LINGKUNGAN

Algoritma Utama

- **Jaringan Sensor IoT:** Algoritma untuk jaringan sensor IoT mengumpulkan dan mengirimkan data lingkungan, menyediakan pemantauan kualitas udara dan air secara waktu nyata. Jaringan ini terdiri dari sensor yang saling terhubung yang mengukur berbagai parameter lingkungan seperti suhu, kelembaban, tingkat polusi, dan indikator kualitas air. Data yang dikumpulkan oleh sensor ini dikirimkan ke sistem terpusat untuk analisis dan pengambilan keputusan. Jaringan sensor IoT menggunakan algoritma untuk memastikan pengumpulan, transmisi, dan pemrosesan data yang efisien. Misalnya, mereka menggunakan algoritma kompresi data untuk meminimalkan penggunaan bandwidth dan konsumsi energi, serta algoritma toleransi kesalahan untuk menjaga integritas data meskipun terjadi kegagalan sensor. Pemantauan waktu nyata melalui sensor IoT memungkinkan deteksi anomali lingkungan secara cepat, memungkinkan tindakan korektif segera untuk mengurangi dampak negatif pada kesehatan masyarakat dan lingkungan.
- **Algoritma Penambangan Data:** Algoritma penambangan data menganalisis data lingkungan untuk mengidentifikasi pola dan memprediksi kondisi lingkungan di masa mendatang. Algoritma ini memproses kumpulan data besar untuk mengungkap pola, korelasi, dan tren tersembunyi yang mungkin tidak langsung terlihat. Teknik seperti pengelompokan, klasifikasi, regresi, dan penambangan aturan asosiasi umumnya digunakan dalam penambangan data lingkungan. Misalnya, algoritma pengelompokan dapat mengelompokkan area dengan tingkat polusi yang serupa, sedangkan algoritma klasifikasi dapat mengkategorikan wilayah berdasarkan risiko bahaya lingkungan. Model regresi dapat memprediksi tingkat polusi di masa depan berdasarkan data



historis, membantu pembuat kebijakan untuk menerapkan langkah-langkah proaktif. Dengan memanfaatkan algoritma penambangan data, ilmuwan lingkungan dan pembuat kebijakan dapat memperoleh wawasan yang lebih dalam tentang masalah lingkungan dan mengembangkan strategi berbasis data untuk pengelolaan sumber daya alam yang berkelanjutan.

Aplikasi

- **Pemantauan Kualitas Udara dan Air:** Menerapkan sensor IoT untuk memantau kualitas udara dan air secara terus menerus, memungkinkan respons tepat waktu terhadap masalah lingkungan.
- **Strategi Lingkungan Berbasis Data:** Menggunakan penambangan data untuk mengembangkan strategi dalam mengatasi tantangan lingkungan berdasarkan pola data yang dianalisis.

Implementasi jaringan sensor IoT untuk pemantauan lingkungan secara waktu nyata, yang secara signifikan meningkatkan pengelolaan kualitas udara dan air. Selain itu, Han dan rekan-rekan. menunjukkan bagaimana algoritma penambangan data dapat menganalisis data lingkungan untuk memprediksi kondisi masa depan dan memberikan informasi untuk strategi proaktif.

19.5 LATIHAN

1. **Sistem Pemungutan Suara Berbasis Blockchain:** Mengembangkan sistem pemungutan suara yang aman dan transparan menggunakan teknologi blockchain untuk meningkatkan kepercayaan dan mencegah kecurangan dalam proses pemilihan. Proyek ini melibatkan perancangan aplikasi terdesentralisasi (DApp) untuk mengelola pendaftaran pemilih, memberikan suara, dan menghitung hasil, memastikan semua transaksi tidak dapat diubah dan dapat diverifikasi.
2. **Chatbot Pemerintah Berbasis NLP:** Buat chatbot cerdas yang menggunakan NLP untuk menangani pertanyaan warga tentang berbagai layanan pemerintah. Chatbot akan membantu pengguna secara waktu nyata, memberikan informasi, membimbing mereka melalui proses aplikasi, dan menangani pertanyaan umum secara otomatis. Hal ini akan meningkatkan aksesibilitas dan mengurangi beban kerja staf pemerintah.
3. **Sistem Manajemen Identitas Digital:** Terapkan sistem verifikasi identitas digital berbasis blockchain untuk menyimpan dan memverifikasi identitas warga secara aman. Sistem ini akan memastikan hanya individu yang berwenang yang dapat mengakses layanan pemerintah yang sensitif, seperti catatan kesehatan dan tunjangan jaminan sosial, sehingga meningkatkan keamanan dan privasi.
4. **Platform Analisis Data Meter Pintar:** Kembangkan platform analisis data untuk menganalisis pola konsumsi dari meter pintar dan memperkirakan permintaan utilitas. Platform ini akan menggunakan algoritma pembelajaran mesin untuk memprediksi konsumsi di masa mendatang, mendeteksi anomali, dan memberikan wawasan untuk mengoptimalkan alokasi sumber daya dan meningkatkan efisiensi energi.



5. **Sistem Respons Permintaan Otomatis:** Rancang dan terapkan sistem respons permintaan otomatis yang menyesuaikan pasokan utilitas berdasarkan data konsumsi waktu nyata. Sistem ini akan memanfaatkan analitik canggih untuk memprediksi periode permintaan puncak dan mengelola distribusi energi, membantu menyeimbangkan pasokan dan permintaan serta mengurangi kelebihan beban jaringan.
6. **Sistem Deteksi Kebocoran yang Didukung IoT:** Membuat sistem yang didukung IoT untuk deteksi dan pelaporan kebocoran air dan gas secara waktu nyata. Proyek ini melibatkan penyebaran sensor pintar yang memantau penggunaan dan mendeteksi anomali, mengirimkan peringatan kepada penyedia utilitas untuk tindakan segera, sehingga mencegah pemborosan sumber daya dan mengurangi biaya.



BAB 20

MANAJEMEN BENCANA

20.1 PENDAHULUAN

Bencana bermanifestasi dalam berbagai bentuk dan secara luas dikategorikan menjadi **bencana alam**, **bencana buatan manusia**, atau **bencana kompleks**. **Bencana antropogenik (buatan manusia)** biasanya diakibatkan oleh kegagalan yang disebabkan oleh manusia, termasuk kecelakaan industri, pelepasan bahan berbahaya, dan kerusakan infrastruktur penting. Sebaliknya, **bencana alam** disebabkan oleh fenomena geofisika, meteorologi, atau hidrologi—seperti gempa bumi, banjir, kekeringan, atau letusan gunung berapi—yang secara signifikan berdampak pada sistem ekologi dan sumber daya alam. **Bencana kompleks**, seperti pandemi dan epidemi, seringkali muncul dari kombinasi faktor biologis, lingkungan, dan sosial ekonomi, sehingga sangat sulit untuk dikelola dan dikendalikan.

Peristiwa seperti itu dapat menimbulkan kerusakan parah pada kehidupan manusia, infrastruktur fisik, ekonomi, dan lingkungan, seringkali melampaui kapasitas respons dan penanggulangan masyarakat yang terkena dampak. Untuk mengatasi tantangan ini, bidang **manajemen risiko bencana (DRM)** telah berkembang sebagai pendekatan berbasis sistem dan interdisipliner yang berfokus pada mitigasi risiko, kesiapan, dan mekanisme respons adaptif. Manajemen Risiko Bencana (DRM) melibatkan koordinasi strategis sumber daya manusia, teknis, dan keuangan untuk meminimalkan dampak bencana. Ini mencakup siklus kegiatan yang komprehensif, termasuk identifikasi bahaya, penilaian risiko, peringatan dini, respons darurat, dan perencanaan pemulihan jangka panjang—yang bertujuan untuk meningkatkan ketahanan dan keberlanjutan populasi dan sistem yang berisiko.

- Untuk menyelidiki desain dan penerapan sistem komunikasi tambang pasca-kecelakaan dan pelacakan personel, dengan fokus pada integrasi arsitektur leaky-feeder yang toleran terhadap kesalahan dan teknologi lokalisasi berbasis identifikasi frekuensi radio (RFID) untuk menjaga ketahanan komunikasi dan kesadaran situasional di lingkungan bawah tanah.
- Untuk mengeksplorasi penerapan teknik penambangan data tingkat lanjut—termasuk ekstraksi informasi terstruktur, peringkasan multi-dokumen, dan pengelompokan spasial-temporal—untuk manajemen informasi bencana yang cerdas, yang memungkinkan pengambilan keputusan secara waktu nyata, deteksi anomali, dan integrasi informasi lintas lembaga.
- Untuk menganalisis peran jaringan sensor nirkabel (WSN) dalam skenario respons bencana, dengan penekanan pada penggunaannya dalam penginderaan lingkungan, lokalisasi berbasis jangkak seluler, dan penyebaran data adaptif untuk koordinasi operasi pencarian, penyelamatan, dan pemulihan yang efisien.



- Untuk mengevaluasi integrasi analitik prediktif, pengelompokan propagasi afinitas, dan algoritma peringkat berbasis grafik leksikal dengan dasbor dinamis untuk peramalan risiko bencana, prioritas sumber daya, dan sistem peringatan dini.
- Untuk meneliti pengintegrasian Algoritma Dunia Cerdas dalam kerangka ketahanan bencana, dengan fokus pada kemampuannya dalam penginderaan otonom, orkestrasi sumber daya cerdas, kolaborasi multi-agen, dan pembelajaran adaptif berbasis AI di perangkat tepi (edge AI), untuk mendukung manajemen bencana skala besar, waktu nyata, dan berbasis konteks di seluruh infrastruktur yang saling terhubung.

Bencana didefinisikan sebagai peristiwa berdampak tinggi—yang terjadi dalam durasi singkat atau lama—yang mengakibatkan kerugian signifikan bagi manusia, infrastruktur, ekonomi, atau lingkungan, melampaui kapasitas penanggulangan komunitas atau sistem yang terkena dampak. Peristiwa ini mengganggu fungsi normal dan membutuhkan bantuan eksternal serta mekanisme respons terkoordinasi untuk pemulihan. Negara berkembang menanggung beban bencana global yang tidak proporsional. Menurut Ibarrarán dan rekan-rekan., lebih dari 95% kematian akibat bencana terjadi di negara berkembang, dan kerugian ekonomi sebagai proporsi produk domestik bruto (PDB) sekitar 20 kali lebih tinggi daripada di negara-negara industri.

Bencana alam didorong oleh proses atau fenomena fisik—seperti gempa bumi, tanah longsor, letusan gunung berapi, badai, tsunami, kebakaran hutan, dan pandemi—yang menyebabkan kerusakan luas pada kehidupan, harta benda, dan ekosistem. Meningkatnya populasi global dan urbanisasi yang dipercepat di wilayah rawan bencana telah meningkatkan frekuensi dan besarnya bencana tersebut. Faktor risiko yang berkontribusi meliputi variabilitas iklim tropis, ketidakstabilan geofisika, penggunaan lahan yang tidak diatur, deforestasi, dan proliferasi infrastruktur non-rekayasa atau informal. Kerangka komunikasi risiko yang tidak memadai, integrasi teknologi yang terbatas, dan program kesiapsiagaan bencana yang kurang didanai memperburuk kerentanan wilayah berkembang. Selain itu, bencana yang **disebabkan oleh manusia (antropogenik)**—yang timbul dari kegagalan teknologi, kesalahan operasional, atau niat jahat—menimbulkan ancaman yang signifikan. Contohnya termasuk kecelakaan penerbangan, ledakan industri, tumpahan limbah beracun, tumpahan minyak, dan tindakan yang disengaja seperti terorisme. Contoh yang menonjol adalah serangan 11 September 2001 di World Trade Center di New York City.

Secara lebih luas, bahaya yang dipicu oleh manusia mencakup peristiwa seperti konflik bersenjata, kerusakan sosial, kebakaran, kegagalan sistem transportasi, insiden radiologis, dan bioterorisme. Beberapa ahli berpendapat bahwa semua bencana pada dasarnya disebabkan oleh manusia, dengan menyebutkan kegagalan dalam tata kelola risiko, perencanaan penggunaan lahan, dan kesiapan darurat sebagai akar penyebab kerentanan. **Kelaparan**, meskipun sering dikaitkan dengan faktor alam seperti kekeringan atau wabah penyakit, semakin diakui sebagai produk dari kegagalan tata kelola sistemik, konflik, atau distribusi sumber daya yang tidak adil. Demikian pula, dampak destruktif gempa bumi diperkuat oleh perencanaan kota yang buruk dan lingkungan binaan yang rentan. Misalnya, risiko tsunami dan



tanah longsor seringkali dapat dikurangi dengan perencanaan pemukiman strategis dan penilaian risiko geospasial.

Bab ini terdiri dari bagian khusus tentang komunikasi tambang pasca kecelakaan dan sistem pelacakan personel, kerangka kerja penambangan data untuk manajemen informasi bencana, dan strategi algoritmik untuk menerapkan jaringan sensor nirkabel cerdas dalam operasi deteksi, respons, dan pemulihan bencana.

20.2 SISTEM DARURAT PASCA-TAMBANG

Bencana pertambangan baru-baru ini di Amerika Serikat telah menyoroti kerentanan kritis dalam infrastruktur keselamatan tambang yang ada, khususnya mengenai komunikasi pasca-insiden dan pelacakan personel. Sebagai tanggapan, Kongres AS memberlakukan Undang-Undang *Peningkatan Tambang dan Tanggap Darurat Baru (MINER)* tahun 2006, yang merupakan revisi paling substansial terhadap peraturan keselamatan tambang dalam lebih dari tiga dekade. Untuk mendukung kepatuhan terhadap Undang-Undang MINER dan untuk mengurangi skenario kehilangan nyawa di masa depan, Institut Nasional untuk Keselamatan dan Kesehatan Kerja (NIOSH) memulai program riset mendalam dan pengembangan ekstensif yang bertujuan untuk memajukan sistem komunikasi bawah tanah. Fokus utama dari upaya ini adalah pengembangan dan peningkatan *sistem komunikasi pasca-kecelakaan*, yang diidentifikasi sebagai titik kegagalan utama dalam beberapa studi kasus bencana. Di antara teknologi yang diusulkan, sistem leaky-feeder yang ditingkatkan muncul sebagai solusi yang layak karena kemampuannya untuk mempertahankan komunikasi dua arah dalam kondisi bawah tanah.

Sistem Leaky-feeder

Sistem leaky-feeder telah lama digunakan di lingkungan terbatas seperti terowongan dan tambang, dengan implementasi dalam operasi penambangan batubara sejak tahun 1980-an. Sistem ini menggunakan kabel feeder koaksial yang berfungsi sebagai saluran komunikasi dan antena terdistribusi. Kabel ini dirancang untuk memancarkan sinyal elektromagnetik melalui perforasi pada pelindung konduktif luarnya, memungkinkan transmisi dan penerimaan sinyal lokal di sepanjang kabel.

Karakteristik Operasional

Sistem leaky-feeder tradisional beroperasi dalam pita frekuensi sangat tinggi (VHF) (30–300 MHz), sedangkan varian yang ditingkatkan beroperasi di dekat 450 MHz dalam rentang frekuensi ultra tinggi (UHF) (300–3000 MHz). Pita frekuensi yang lebih tinggi memfasilitasi peningkatan bandwidth dan peningkatan throughput data, yang menguntungkan untuk komunikasi dan telemetri waktu nyata. Selain itu, sinyal UHF menunjukkan karakteristik propagasi yang unggul dalam geometri tambang yang kompleks, khususnya di sekitar potongan melintang dan sudut pilar. Namun, manfaat ini disertai dengan konsekuensi negatif, termasuk tingkat atenuasi yang lebih tinggi dan biaya infrastruktur yang lebih besar karena persyaratan perangkat keras yang lebih ketat.



Topologi Sistem dan Arsitektur Daya

Untuk mematuhi standar izin, infrastruktur komunikasi dibagi menjadi *sel-sel* terisolasi, masing-masing dibatasi oleh penghalang elektromagnetik yang memungkinkan transmisi sinyal RF sambil mengisolasi distribusi daya DC. Setiap sel ditenagai oleh catu daya independen, dan integritas sinyal dipertahankan melalui penggunaan penguat dua arah, sejajar dengan kontrol penguatan otomatis. Penguat ini biasanya ditempatkan dengan jarak sekitar 1000 kaki untuk mengkompensasi kehilangan saluran dan memastikan cakupan sinyal yang berkelanjutan.

Teknik Perluasan Cakupan

Jangkauan komunikasi sistem leaky-feeder standar sebagian besar terbatas pada terowongan utama (pintu masuk tambang) yang menampung kabel pengumpan, dengan perluasan terbatas ke potongan melintang lateral. Untuk memperluas jangkauan operasional, arsitektur sistem canggih menggabungkan splitter RF rugi rendah yang bercabang dari kabel utama dan berinteraksi dengan antena yang ditempatkan secara strategis di koridor terbuka tanpa peralatan. Konfigurasi ini meningkatkan jangkauan sinyal sekaligus meminimalkan interferensi propagasi.

Ketahanan dan Redundansi Sistem

Tujuan fungsional inti dari sistem komunikasi tambang adalah untuk memastikan konektivitas pasca-insiden antara personel bawah tanah yang terjebak dan pusat komando di atas tanah. Akibatnya, ketahanan sistem dalam kondisi ekstrem—seperti ledakan, runtuhnya atap, kebakaran, dan banjir—sangat penting. Untuk mencapai hal ini diperlukan penguatan komponen dan redundansi komunikasi. Dalam satu implementasi yang disponsori NIOSH, sistem leaky-feeder skala penuh dipasang di sepanjang bagian tambang batubara West Virginia sepanjang 8 mil. Redundansi diwujudkan melalui jalur komunikasi sekunder yang dibangun melalui tautan serat optik antara poros udara tambahan dan stasiun pangkalan utama yang terletak di poros lift.

Subsistem Pelacakan Personel

Sesuai dengan Undang-Undang MINER, rencana tanggap darurat tambang harus memasukkan mekanisme untuk lokalisasi pekerja bawah tanah secara waktu nyata. Karena sistem leaky-feeder saja tidak dapat mendukung fungsi ini, diperlukan subsistem pelacakan khusus. Solusi yang banyak diadopsi melibatkan penggunaan teknologi RFID. Sistem ini terdiri dari pembaca RFID yang terdistribusi secara spasial—masing-masing diberi pengenalan posisi unik—dan tag RFID aktif atau pasif yang ditempelkan pada helm atau perlengkapan penambang. Saat penambang melintasi tambang, tag mereka menyiarkan pengenalan unik ke pembaca terdekat. Pembaca kemudian mengirimkan ID tag dan lokasinya sendiri ke pusat kendali permukaan melalui backhaul data yang tersedia—biasanya melalui saluran leaky-feeder, kabel twisted-pair, serat optik, atau node relai nirkabel—sehingga memungkinkan pelacakan personel secara waktu nyata dan kesadaran situasional selama keadaan darurat.



20.3 MANAJEMEN INFORMASI BENCANA

Sistem manajemen informasi bencana (DIM) memainkan peran penting dalam mendukung pengambilan keputusan berbasis data secara waktu nyata selama keadaan darurat. Domain seperti respons terhadap badai telah menyoroti tantangan kritis: ketidakstabilan infrastruktur komunikasi mengganggu aliran informasi yang konsisten, dan volume data heterogen yang sangat besar menghambat asimilasi tepat waktu dan koordinasi respons. Untuk mengurangi tantangan ini, diusulkan arsitektur sistem yang kuat yang mengintegrasikan penambahan data tingkat lanjut, pemrosesan bahasa alami (NLP), dan analitik geospasial. Sistem ini dimulai dengan modul ekstraksi informasi terstruktur yang menerapkan teknik Pengenalan Entitas Bernama (NER) berbasis pembelajaran mendalam seperti BiLSTM-CRF dan model berbasis transformer seperti BERT dan RoBERTa untuk mengidentifikasi entitas kunci—termasuk lokasi, fasilitas, status sumber daya, dan ekspresi waktu. Setelah diekstrak, entitas dan penanda temporal ini diklasifikasikan ke dalam kategori semantik (misalnya, terbuka, tertutup, tidak pasti) menggunakan kerangka klasifikasi berbasis rentang dan model bahasa yang disempurnakan.

Untuk mengompresi dan meringkas beberapa laporan, strategi peringkasan multi-dokumen diimplementasikan berdasarkan algoritma propagasi afinitas (AP). Metode ini membangun grafik kesamaan kalimat menggunakan penyematan kontekstual yang dihasilkan oleh model seperti Sentence-BERT atau Universal Sentence Encoder, dengan tepi yang diberi bobot berdasarkan kesamaan kosinus. Melalui penyampaian pesan iteratif, algoritma AP menentukan kalimat contoh berdasarkan metrik tanggung jawab dan ketersediaan, secara rekursif membentuk struktur ringkasan hierarkis. Pemilihan kalimat selanjutnya disempurnakan menggunakan mekanisme penilaian yang menggabungkan kemungkinan model bahasa (misalnya, GPT-2), sentralitas LexPageRank, dan skor kesegaran temporal. Untuk menghilangkan konten yang berlebihan dan meningkatkan relevansi, digunakan mesin deduplikasi dan rekomendasi konten. Metode hashing semantik seperti SimHash dan MinHash digunakan untuk deteksi duplikat yang hampir sama, sementara pengelompokan aglomeratif berdasarkan TF-IDF dan kesamaan Jaccard memastikan pengelompokan konten yang serupa. Sistem rekomendasi neural berdasarkan penyaringan kolaboratif yang digerakkan oleh perhatian (misalnya, penyaringan kolaboratif neural, NCF) memberi peringkat fragmen laporan sesuai dengan peran pengguna, interaksi historis, dan konteks kueri saat ini. Dasbor front-end diperbarui secara dinamis melalui API GraphQL dan mendukung visualisasi yang difilter melalui formulir kueri adaptif. Pembaruan penting diberi peringkat menggunakan algoritma pembelajaran-untuk-peringkat seperti LambdaMART, mengoptimalkan visibilitas wawasan prioritas tinggi.

Analisis data geospasial membentuk lapisan kunci dalam kesadaran situasional bencana. Algoritma pengelompokan spasial seperti DBSCAN dan varian hierarkisnya HDBSCAN digunakan untuk mengidentifikasi wilayah padat kerusakan infrastruktur atau konsentrasi bahaya. Aset dan peristiwa yang diberi geotag juga dimodelkan sebagai grafik spasial, memungkinkan deteksi komunitas menggunakan teknik berbasis modularitas seperti algoritma Louvain atau Leiden. Untuk mendeteksi pola peristiwa spasial-temporal yang



berulang, sistem ini mengintegrasikan penambangan urutan berbasis apriori yang disesuaikan untuk dataset temporal yang diberi geotag. Dengan menggabungkan modul cerdas ini, kerangka kerja DIM menjadi mampu memberikan wawasan yang kaya secara semantik, non-redundan, dan kontekstual secara geospasial. Hal ini meningkatkan efisiensi perencanaan respons bencana, mengoptimalkan alokasi sumber daya, dan memberikan pemangku kepentingan operasional gambaran situasional yang komprehensif dan waktu nyata dalam kondisi dinamis dan tidak pasti.

20.4 ALGORITMA SENSOR KEBENCANAAN

WSN mewakili paradigma siber-fisik canggih yang menggabungkan sistem tertanam, protokol komunikasi nirkabel, dan teknologi penginderaan terdistribusi untuk memungkinkan pemantauan dan pengendalian lingkungan skala besar dan waktu nyata. Jaringan sensor nirkabel (WSN) tipikal terdiri dari banyak node sensor berbiaya rendah dan berdaya rendah yang terdistribusi secara spasial yang berkolaborasi untuk melakukan tugas penginderaan, pemrosesan, dan komunikasi terdistribusi secara otonom. Karena sifatnya yang mengatur diri sendiri dan skalabilitasnya, WSN sangat cocok untuk penyebaran di lingkungan rawan bencana atau tidak dapat diakses di mana infrastruktur komunikasi konvensional tidak tersedia atau terganggu.

Dalam konteks manajemen bencana, WSN berperan penting dalam mendukung aplikasi yang kritis waktu seperti pemantauan aktivitas seismik, penilaian kesehatan struktural, deteksi kebakaran hutan, deteksi kebocoran bahan kimia, dan operasi pencarian dan penyelamatan pasca bencana. Jaringan ini memfasilitasi sistem peringatan dini dan kesadaran situasional dengan mengumpulkan data spasial-temporal dari zona berbahaya dan meneruskannya ke unit komando terpusat atau node komputasi tepi. Persyaratan utama untuk fungsionalitas WSN dalam skenario tersebut adalah lokalisasi node, karena pengetahuan tentang lokasi geografis peristiwa atau pembacaan sensor sangat penting untuk interpretasi dan respons yang bermakna.

Algoritma lokalisasi untuk WSN dapat diklasifikasikan secara luas menjadi metode berbasis jangkauan dan tanpa jangkauan. Lokalisasi berbasis jangkauan biasanya menggunakan metrik fisik seperti indikator kekuatan sinyal yang diterima (RSSI), waktu kedatangan (ToA), atau sudut kedatangan (AoA) untuk memperkirakan jarak antara node sensor dan jangkar referensi. Di antara teknik-teknik ini, teknik berbasis RSSI lebih disukai di lingkungan bencana yang terbatas sumber daya karena kesederhanaan dan kelayakan perangkat kerasnya. Untuk meningkatkan presisi, lokalisasi sering memanfaatkan node jangkar bergerak yang melintasi area penyebaran pada lintasan yang telah ditentukan dan secara berkala menyiarkan sinyal suara di titik arah yang ditentukan. Node yang menerima suara ini memperkirakan posisinya melalui trilaterasi atau multilaterasi, berdasarkan konfigurasi geometris posisi jangkar yang diketahui dan kekuatan sinyal paket yang diterima.

Lokalisasi Berbasis RSSI dengan Jangkar Bergerak

Algoritma berikut menguraikan alur kerja tipikal untuk lokalisasi berbasis RSSI menggunakan node jangkar bergerak di dalam WSN zona bencana:



Algoritma 5: Trilaterasi Berbasis RSSI dengan Jangkar Bergerak

1. **Masukan:** Trajektori jangkar yang telah ditentukan dan interval transmisi; ambang batas RSSI τ
2. Tentukan jangkauan komunikasi r untuk node jangkar bergerak berdasarkan medan dan model propagasi.
3. $\forall$ posisi suar yang telah ditentukan di sepanjang trajektori jangkar.
4. Siarkan pesan suar yang memuat koordinat jangkar.
5. AKHIR UNTUK
6. $\forall$ node sensor tak dikenal di sekitarnya.
7. Terima beberapa pesan suar dan catat nilai RSSI.
8. Perkirakan jarak relatif d_i dari RSSI menggunakan model kehilangan jalur.
9. Periksa apakah tersedia tiga atau lebih titik suar yang tidak segaris.
10. **JIKA** ya
11. Terapkan algoritma trilaterasi untuk menghitung perkiraan posisi (x, y) .
12. **JIKA** TIDAK
13. Tunggu pesan suar tambahan atau pindah ke area cakupan baru.
14. **AKHIR** **JIKA**
15. **AKHIR** **UNTUK**

Model kehilangan jalur yang digunakan untuk mengubah nilai RSSI menjadi estimasi jarak biasanya dirumuskan sebagai berikut:

$$\text{RSSI}(d) = \text{RSSI}(d_0) - 10n\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma$$

di mana d adalah jarak dari jangkar, d_0 adalah jarak referensi, n adalah eksponen kehilangan jalur, dan X_σ adalah variabel acak Gaussian yang mewakili pelemahan sinyal.

Studi terbaru juga telah mengeksplorasi integrasi pendekatan pembelajaran mesin, seperti Regresi Proses Gaussian (GPR) dan jaringan saraf dalam (DNN), untuk meningkatkan akurasi lokalisasi di lingkungan pasca-bencana yang tidak memiliki garis pandang (NLoS) atau sangat padat. Selain itu, model hibrida yang menggabungkan penginderaan inersia dengan lokalisasi berbasis radio muncul untuk mendukung pelacakan mobilitas dalam operasi penyelamatan dinamis. WSN, ketika disematkan dengan algoritma lokalisasi cerdas, menyediakan infrastruktur yang tangguh untuk penilaian situasi otonom dan dukungan pengambilan keputusan di lingkungan yang dilanda bencana. Sinergi mereka dengan robot seluler, UAV, dan sistem AI tepi semakin meningkatkan potensi mereka dalam aplikasi Smart World untuk pemantauan infrastruktur yang tangguh dan respons darurat.

20.5 LATIHAN

1. **Sistem Komunikasi Tambang Pasca Kecelakaan yang Ditingkatkan**



Untuk mengembangkan dan mengimplementasikan sistem komunikasi leaky-feeder canggih di tambang untuk memastikan komunikasi pasca kecelakaan dan pelacakan penambang yang efektif.

Deskripsi: Proyek ini melibatkan perancangan dan penerapan sistem leaky-feeder dengan fitur frekuensi dan redundansi yang ditingkatkan. Ini akan mencakup pengembangan antarmuka pemantauan waktu nyata yang dapat digunakan oleh personel di atas tanah untuk berkomunikasi dengan penambang dan melacak lokasi mereka menggunakan teknologi RFID.

2. **Penambangan Data untuk Manajemen Informasi Bencana Waktu Nyata**

Untuk membuat sistem penambangan data yang mengekstrak, memproses, dan meringkas informasi penting dari berbagai laporan selama bencana.

Deskripsi: Proyek ini akan berfokus pada pengembangan arsitektur sistem yang menggunakan metode ekstraksi informasi dan teknik peringkasan multi-dokumen hierarkis. Sistem ini akan menghasilkan dasbor dinamis waktu nyata yang menampilkan informasi bencana penting yang tidak redundan untuk para pengambil keputusan.

3. **Jaringan Sensor Cerdas untuk Operasi Bantuan Bencana**

Untuk menerapkan WSN untuk operasi pencarian dan penyelamatan yang efisien di daerah yang terkena bencana.

Deskripsi: Proyek ini akan melibatkan pengaturan WSN (Wireless Sensor Network) dengan node sensor berbiaya rendah yang mampu mendeteksi tanda-tanda kehidupan dan kondisi lingkungan. Proyek ini akan mencakup pengembangan algoritma untuk lokalisasi node sensor dan mengintegrasikan sistem komunikasi untuk mengirimkan data ke tim penyelamat.

4. **Deteksi Dini dan Manajemen Pandemi Menggunakan Pembelajaran Mesin**

Untuk mengembangkan model pembelajaran mesin untuk deteksi dini dan manajemen pandemi seperti Coronavirus.

Deskripsi: Proyek ini akan melibatkan pengumpulan dan analisis data kesehatan untuk melatih model pembelajaran mesin yang dapat memprediksi potensi wabah. Model ini akan diintegrasikan dengan sistem manajemen kesehatan masyarakat untuk memberikan peringatan dan rekomendasi tepat waktu untuk tindakan penahanan.

5. **Manajemen Risiko Gempa Bumi untuk Industri Pariwisata**

Untuk menciptakan mekanisme praktis untuk penilaian dan manajemen risiko bencana gempa bumi yang disesuaikan dengan industri pariwisata.

Deskripsi: Proyek ini akan menggabungkan ilmu bumi, teknik, dan data asuransi untuk mengembangkan sistem manajemen risiko yang komprehensif. Ini akan berfokus pada strategi pencegahan, perencanaan asuransi, dan alokasi sumber daya untuk mengurangi dampak gempa bumi pada sektor pariwisata.

6. **Dasbor Dinamis Real-Time untuk Koordinasi Respons Bencana**

Untuk mengembangkan dasbor dinamis waktu nyata untuk mengkoordinasikan upaya respons bencana.



Deskripsi: Proyek ini akan melibatkan pembuatan dasbor yang didukung oleh mesin rekomendasi konten yang mengekstrak dan menampilkan informasi yang paling relevan dan tidak berlebihan dari berbagai sumber. Dasbor ini akan membantu tim manajemen bencana untuk dengan cepat menilai situasi dan membuat keputusan yang tepat.

7. **Sistem Pemulihan Bencana dan Alokasi Sumber Daya Berbasis Komunitas**

Untuk merancang sistem yang mengidentifikasi dan mendukung upaya pemulihan bencana berbasis komunitas.

Deskripsi: Proyek ini akan menggunakan algoritma pengelompokan spasial untuk mengidentifikasi komunitas yang terkena dampak bencana. Sistem ini akan memfasilitasi alokasi sumber daya dan koordinasi di antara berbagai pemangku kepentingan, memastikan bahwa daerah yang paling terkena dampak menerima dukungan yang tepat waktu dan memadai.



BAB 21

ALGORITMA KOMUNIKASI

21.1 ALGORITMA KOMUNIKASI UNTUK WSN

Dalam konteks kota pintar dan sistem dunia pintar, algoritma komunikasi memainkan peran penting dalam memastikan transfer data yang efisien dan andal di antara perangkat dan sensor yang saling terhubung. Algoritma deteksi dan koreksi kesalahan, seperti pemeriksaan redundansi siklik (CRC) dan kode Hamming, menjaga integritas data di seluruh jaringan yang luas. Algoritma perutean seperti vektor jarak, perutean status tautan, dan vektor jarak sesuai permintaan ad hoc (AODV) mengoptimalkan jalur data di lingkungan dinamis, memastikan respons waktu nyata. Algoritma kontrol aliran, termasuk stop-and-wait dan sliding window, mengatur aliran data untuk mencegah kemacetan, sehingga memungkinkan interaksi dan koordinasi yang lancar di antara berbagai komponen kota pintar. Algoritma-algoritma ini secara kolektif meningkatkan kinerja, keandalan, dan skalabilitas infrastruktur kota pintar, mendukung aplikasi canggih seperti manajemen lalu lintas, pemantauan lingkungan, dan sistem keselamatan publik. Tujuan bab ini adalah sebagai berikut:

- Untuk mengeksplorasi algoritma komunikasi untuk jaringan sensor nirkabel (WSN) dan perannya dalam mengoptimalkan propagasi data dan efisiensi energi.
- Untuk menganalisis algoritma komunikasi simpan-bawa-teruskan untuk jaringan toleran tunda (DTN), dengan fokus pada penerapannya di lingkungan konektivitas intermiten.
- Untuk memeriksa algoritma komunikasi daya rendah untuk jaringan daya rendah dan rentan kehilangan data (LLN) untuk mengoptimalkan konsumsi energi dan meningkatkan kinerja jaringan.
- Untuk menyelidiki algoritma jaringan yang ditentukan perangkat lunak (SDN) dan kontribusinya terhadap kontrol terpusat dan manajemen sumber daya jaringan dinamis.
- Untuk menilai algoritma jaringan peer-to-peer (P2P) dan efektivitasnya dalam jaringan berbagi data terdistribusi dan terdesentralisasi.

Algoritma komunikasi sangat penting untuk mengoptimalkan WSN dan penyebaran data. Algoritma ini memastikan komunikasi yang efisien dan stabilitas jaringan dengan memilih subset node, yang dikenal sebagai Himpunan Dominasi, untuk mengelola tugas komunikasi. Tujuan utamanya adalah untuk meminimalkan konsumsi energi dan memperpanjang umur jaringan.

Algoritma Utama

- **Himpunan Dominasi Terhubung Minimum (MCDS):** Algoritma ini memilih jumlah node minimal sehingga setiap node dalam jaringan merupakan bagian dari MCDS atau terhubung langsung ke node dalam MCDS. Teknik seperti algoritma greedy dan metode berbasis pohon umumnya digunakan untuk mencapai hal ini.



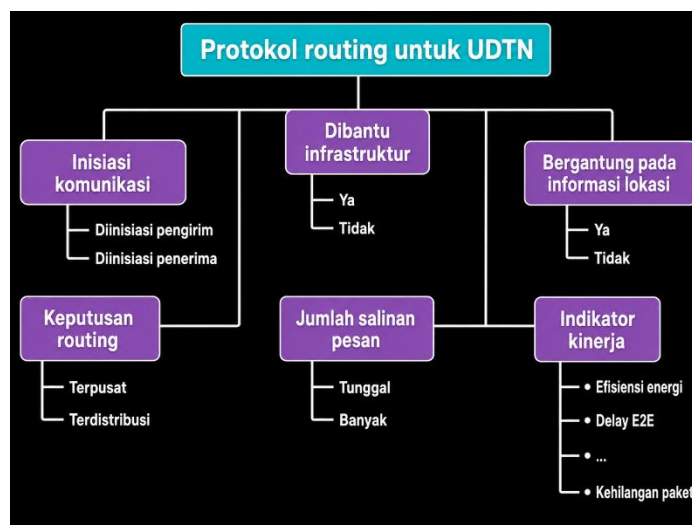
- **Algoritma Berbasis Klaster:** Algoritma ini mengelompokkan node ke dalam klaster dengan kepala klaster bertindak sebagai koordinator. Pembentukan dan pemeliharaan klaster mengurangi overhead komunikasi dan meningkatkan efisiensi energi.
- **Konstruksi Jaringan Tulang Punggung Virtual:** Ini melibatkan pembuatan jaringan tulang punggung virtual untuk memfasilitasi perutean data yang efisien dan mengurangi konsumsi energi.

Aplikasi

Dalam aplikasi praktis, algoritma MCDS digunakan dalam sistem pemantauan lingkungan di mana WSN melacak berbagai parameter seperti suhu, kelembaban, dan tingkat polusi. Dengan mengoptimalkan jalur komunikasi, algoritma ini mengurangi penggunaan energi, memastikan bahwa node sensor tetap berfungsi untuk jangka waktu yang lama, sehingga meningkatkan keandalan dan efisiensi pemantauan lingkungan.

21.2 DTN: ALGORITMA SIMPAN-BAWA-TERUSKAN

Algoritma simpan-bawa-teruskan sangat penting dalam DTN, di mana konektivitas jaringan bersifat intermiten. Algoritma ini memungkinkan paket data untuk disimpan dan dibawa oleh node hingga muncul peluang penerusan yang sesuai, memastikan bahwa data akhirnya mencapai tujuannya (Gambar 21.1).



Gambar 21.1 Protokol Perutean untuk DTN.

Algoritma Utama

- **Perutean Epidemi:** Node terus menerus mereplikasi dan meneruskan pesan ke node yang ditemui, mirip dengan penyebaran epidemi. Hal ini meningkatkan kemungkinan pengiriman pesan dalam kondisi jaringan yang jarang dan sangat dinamis.
- **Semprot dan Tunggu:** Algoritma ini menyemprotkan sejumlah salinan pesan ke dalam jaringan dan menunggu salah satu salinan mencapai tujuan, menyeimbangkan antara penggunaan sumber daya dan probabilitas pengiriman.



- **PROPHET (Protokol Perutean Probabilistik Menggunakan Riwayat Pertemuan dan Transitivitas):** Algoritma ini menggunakan riwayat pertemuan dan transitivitas untuk memprediksi probabilitas keberhasilan pengiriman pesan ke tujuan.

Aplikasi

Dalam aplikasi dunia nyata, seperti pelacakan satwa liar dan penginderaan jauh di lingkungan yang keras, metode simpan-bawa-teruskan memastikan transmisi data yang andal meskipun sering terjadi gangguan jaringan. Teknik ini sangat penting untuk menjaga komunikasi dalam skenario di mana jaringan tradisional gagal karena kurangnya konektivitas berkelanjutan.

21.3 ALGORITMA KOMUNIKASI HEMAT ENERGI LLN

LLN membutuhkan algoritma komunikasi yang mengoptimalkan konsumsi daya karena sumber daya energi node yang terbatas. Algoritma ini berfokus pada meminimalkan penggunaan energi sambil mempertahankan komunikasi yang efektif.

Algoritma Utama

- **Protokol Perutean untuk Jaringan Daya Rendah dan Rugi (RPL):** RPL membangun Tujuan Grafik Acyclic Terarah Berorientasi Tujuan (DODAG) berdasarkan fungsi tujuan seperti meminimalkan konsumsi energi dan memaksimalkan keandalan.
- **Hierarki Pengelompokan Adaptif Energi Rendah (LEACH):** LEACH adalah protokol hierarkis yang mengatur node ke dalam cluster untuk meminimalkan disipasi energi dalam jaringan sensor.
- **Time Slotted Channel Hopping (TSCH):** TSCH adalah protokol MAC yang menggunakan slot waktu untuk menyinkronkan node dan channel hopping untuk mengurangi interferensi, meningkatkan efisiensi energi dan keandalan.

Aplikasi

RPL banyak digunakan di rumah pintar dan sistem otomatisasi industri. Ini memungkinkan perangkat dengan daya dan sumber daya komputasi yang terbatas untuk berkomunikasi secara efisien, memastikan operasi jaringan jangka panjang dan mengurangi kebutuhan penggantian baterai yang sering.

21.4 ALGORITMA KONTROL JARINGAN SDN

SDN memisahkan bidang kontrol dari bidang data, memungkinkan manajemen jaringan yang lebih fleksibel dan terprogram. Algoritma SDN memungkinkan kontrol terpusat dan optimasi sumber daya jaringan.

Algoritma Utama

- **OpenFlow:** OpenFlow menyediakan antarmuka standar untuk memprogram tabel aliran sakelar jaringan, memungkinkan pengontrol SDN pusat untuk secara dinamis mengelola dan mengoptimalkan aliran lalu lintas di seluruh jaringan.
- **Open Network Operating System (ONOS):** ONOS adalah sistem operasi SDN yang dirancang untuk ketersediaan tinggi, kinerja, dan skalabilitas dalam jaringan operator.



- **P4 (Programming Protocol-independent Packet Processors):** P4 adalah bahasa tingkat tinggi yang dirancang untuk memprogram prosesor paket yang independen dari protokol, memungkinkan operasi jaringan yang lebih fleksibel dan efisien.

Aplikasi

Di pusat data dan jaringan perusahaan besar, SDN dan OpenFlow digunakan untuk meningkatkan kelincuhan dan efisiensi jaringan. Mereka memungkinkan penyeimbangan beban dinamis, manajemen lalu lintas yang lebih baik, dan penyebaran layanan baru yang lebih cepat, secara signifikan meningkatkan kinerja jaringan secara keseluruhan.

21.5 ALGORITMA JARINGAN PEER-TO-PEER

Algoritma jaringan P2P memfasilitasi berbagi data langsung antar node tanpa bergantung pada server terpusat. Algoritma ini sangat penting untuk jaringan terdistribusi di mana desentralisasi merupakan persyaratan utama.

Algoritma Utama

- **Protokol BitTorrent:** BitTorrent memecah file menjadi bagian-bagian yang lebih kecil, memungkinkan pengguna untuk mengunduh bagian-bagian dari beberapa peer secara bersamaan, meningkatkan kecepatan unduhan dan mengurangi beban pada setiap peer.
- **Chord:** Chord adalah protokol pencarian P2P yang dapat diskalakan yang menyediakan tabel hash terdistribusi untuk lokasi node dan data yang efisien.
- **Kademlia:** Kademlia adalah algoritma jaringan P2P terdesentralisasi yang menggunakan tabel hash terdistribusi untuk pencarian node dan penyimpanan data yang efisien.

Aplikasi

BitTorrent dan algoritma P2P serupa digunakan untuk berbagi file, distribusi konten, dan proyek kolaboratif. Algoritma ini sangat efektif untuk mendistribusikan file besar dengan cepat dan efisien, sehingga populer untuk berbagi perangkat lunak, media, dan kumpulan data besar.

21.6 LATIHAN

1. Implementasikan dan optimalkan algoritma MCDS, berbasis kluster, dan konstruksi tulang punggung virtual. Evaluasi kinerja algoritma ini dalam hal efisiensi energi, masa pakai jaringan, dan overhead komunikasi.
2. Implementasikan dan evaluasi algoritma Epidemic Routing, Spray and Wait, dan PROPHET dalam lingkungan DTN simulasi. Analisis kinerja algoritma ini dalam hal rasio pengiriman, latensi, dan penggunaan sumber daya. Kembangkan peningkatan pada algoritma yang ada untuk meningkatkan efisiensi dan keandalannya.
3. Implementasikan protokol RPL, LEACH, dan TSCH untuk LLN.
4. Evaluasi kinerja protokol ini dalam hal konsumsi energi, keandalan jaringan, dan skalabilitas. Implementasikan dan evaluasi OpenFlow, ONOS, dan P4 dalam lingkungan



jaringan simulasi. Analisis dampaknya terhadap fleksibilitas jaringan, kinerja, dan manajemen sumber daya.

5. Implementasikan dan evaluasi algoritma BitTorrent, Chord, dan Kademlia dalam jaringan P2P simulasi. Analisis kinerjanya dalam hal efisiensi distribusi data, skalabilitas jaringan, dan toleransi kesalahan.

6. Deteksi Dini dan Manajemen Epidemik Menggunakan Pembelajaran Mesin

Tujuan: Mengembangkan model pembelajaran mesin untuk deteksi dini dan manajemen epidemik seperti Coronavirus.

Deskripsi: Proyek ini akan melibatkan pengumpulan dan analisis data kesehatan untuk melatih model pembelajaran mesin yang dapat memprediksi potensi wabah. Model ini akan diintegrasikan dengan sistem manajemen kesehatan masyarakat untuk memberikan peringatan dan rekomendasi tepat waktu untuk tindakan penahanan.

7. Manajemen Risiko Gempa Bumi untuk Industri Pariwisata

Tujuan: Menciptakan mekanisme praktis untuk penilaian dan manajemen risiko bencana gempa bumi yang disesuaikan dengan industri pariwisata.

Deskripsi: Proyek ini akan menggabungkan ilmu bumi, teknik, dan data asuransi untuk mengembangkan sistem manajemen risiko yang komprehensif. Proyek ini akan berfokus pada strategi pencegahan, perencanaan asuransi, dan alokasi sumber daya untuk mengurangi dampak gempa bumi pada sektor pariwisata.

8. Dasbor Dinamis Waktu Nyata untuk Koordinasi Respons Bencana

Tujuan: Mengembangkan dasbor dinamis waktu nyata untuk mengkoordinasikan upaya respons bencana.

Deskripsi: Proyek ini akan melibatkan pembuatan dasbor yang didukung oleh mesin rekomendasi konten yang mengekstrak dan menampilkan informasi yang paling relevan dan tidak berlebihan dari berbagai sumber. Dasbor ini akan membantu tim manajemen bencana untuk dengan cepat menilai situasi dan membuat keputusan yang tepat.

9. Sistem Pemulihan Bencana dan Alokasi Sumber Daya Berbasis Komunitas

Tujuan: Merancang sistem yang mengidentifikasi dan mendukung upaya pemulihan bencana berbasis komunitas.

Deskripsi: Proyek ini akan menggunakan algoritma pengelompokan spasial untuk mengidentifikasi komunitas yang terkena dampak bencana. Sistem ini akan memfasilitasi alokasi sumber daya dan koordinasi di antara berbagai pemangku kepentingan, memastikan bahwa daerah yang paling terkena dampak menerima dukungan yang tepat waktu dan memadai.



DAFTAR PUSTAKA

- Alsharif, M. H., Kelechi, A. H., Yahya, K., & Chaudhry, S. A. (2020). Machine learning algorithms for smart data analysis in internet of things environment: taxonomies and research trends. *Symmetry*, 12(1), 88.
- Bathla, G., Bhadane, K., Singh, R. K., Kumar, R., Aluvalu, R., Krishnamurthi, R.,... & Basheer, S. (2022). Autonomous vehicles and intelligent automation: Applications, challenges, and opportunities. *Mobile information systems*, 2022(1), 7632892.
- Belgaum, M. R., Alansari, Z., Musa, S., Alam, M. M., & Mazliham, M. S. (2021). Role of artificial intelligence in cloud computing, IoT and SDN: Reliability and scalability issues. *International Journal of Electrical & Computer Engineering (2088-8708)*, 11(5), 4458.
- Bi, S., Wang, C., Zhang, J., Huang, W., Wu, B., Gong, Y., & Ni, W. (2022). A survey on artificial intelligence aided internet-of-things technologies in emerging smart libraries. *Sensors*, 22(8), 2991.
- Chander, B., Pal, S., De, D., & Buyya, R. (2022). Artificial intelligence-based internet of things for industry 5.0. In *Artificial intelligence-based internet of things systems* (pp. 3-45). Cham: Springer International Publishing.
- Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0. *Sensors*, 23(16), 7194.
- Choudhary, A. (2024). Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. *Discover Internet of Things*, 4(1), 31.
- Datta, S. D., Islam, M., Sobuz, M. H. R., Ahmed, S., & Kar, M. (2024). Artificial intelligence and machine learning applications in the project lifecycle of the construction industry: A comprehensive review. *Heliyon*, 10(5).
- Dey, N., Mahalle, P. N., Shafi, P. M., Kimabahune, V. V., & Hassanien, A. E. (Eds.). (2020). *Internet of Things, smart computing and technology: a roadmap ahead*. Springer International Publishing.
- Elahi, M., Afolaranmi, S. O., Martinez Lastra, J. L., & Perez Garcia, J. A. (2023). A comprehensive literature review of the applications of AI techniques through the lifecycle of industrial equipment. *Discover artificial intelligence*, 3(1), 43.



- Esenogho, E., Djouani, K., & Kurien, A. M. (2022). Integrating artificial intelligence Internet of Things and 5G for next-generation smartgrid: A survey of trends challenges and prospect. *Ieee Access*, 10, 4794-4831.
- Jararweh, Y., Fatima, S., Jarrah, M., & AlZu'bi, S. (2023). Smart and sustainable agriculture: Fundamentals, enabling technologies, and future directions. *Computers and Electrical Engineering*, 110, 108799.
- Jhaveri, R. H., Revathi, A., Ramana, K., Raut, R., & Dhanaraj, R. K. (2022). A review on machine learning strategies for real-world engineering applications. *Mobile Information Systems*, 2022(1), 1833507.
- Kapoor, A. (2025). *Hands-On Artificial Intelligence for IoT: Expert machine learning and deep learning techniques for developing smarter IoT systems*. Packt Publishing Ltd.
- Khan, I. U., Ouaisa, M., Ouaisa, M., Fayaz, M., & Ullah, R. (Eds.). (2024). Artificial intelligence for intelligent systems: fundamentals, challenges, and applications.
- Khang, A., Shah, V., & Rani, S. (Eds.). (2023). *Handbook of Research on AI-Based Technologies and Applications in the Era of the Metaverse*. IGI Global.
- Krishnan, R. B., & Kumar, N. R. (Eds.). (2026). *Quantum Machine Learning: Artificial Intelligence for Smart Internet of Things Applications*. John Wiley & Sons.
- Mahmood, M. R., Matin, M. A., Sarigiannidis, P., & Goudos, S. K. (2022). A comprehensive review on artificial intelligence/machine learning algorithms for empowering the future IoT toward 6G era. *IEEE access*, 10, 87535-87562.
- Mathew, D., Brintha, N. C., & Jappes, J. W. (2023). Artificial intelligence powered automation for industry 4.0. In *New horizons for Industry 4.0 in modern business* (pp. 1-28). Cham: Springer International Publishing.
- Mazhar, T., Irfan, H. M., Haq, I., Ullah, I., Ashraf, M., Shloul, T. A.,... & Elkamchouchi, D. H. (2023). Analysis of challenges and solutions of IoT in smart grids using AI and machine learning techniques: A review. *Electronics*, 12(1), 242.
- Mowla, M. N., Mowla, N., Shah, A. S., Rabie, K. M., & Shongwe, T. (2023). Internet of Things and wireless sensor networks for smart agriculture applications: A survey. *IEEE Access*, 11, 145813-145852.
- Qureshi, K. N., & Newe, T. (Eds.). (2024). *Artificial intelligence of things (AIoT): new standards, technologies and communication systems*. CRC Press.



- Rathor, A. S., Choudhury, S., Sharma, A., Nautiyal, P., & Shah, G. (2024). Empowering vertical farming through IoT and AI-Driven technologies: A comprehensive review. *Heliyon*, 10(15).
- Ryalat, M., Almtireen, N., Al-refai, G., Elmoaqet, H., & Rawashdeh, N. (2025). Research and education in robotics: A comprehensive review, trends, challenges, and future directions. *Journal of Sensor and Actuator Networks*, 14(4), 76.
- Saha, S., & Shukla, S. (2026). *Algorithms for Smart World Technologies: A Comprehensive Guide to Applications in AI, IoT and Automation for Electrical and Computer Engineers*. John Wiley & Sons.
- Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2023). Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile Networks and Applications*, 28(1), 296-312.
- Tabassum, S., Vijay Babu, A. R., & Dheer, D. K. (2024). A comprehensive exploration of IoT-enabled smart grid systems: power quality issues, solutions, and challenges. *Science and technology for energy transition*, 79, 62.
- Taj, I., & Zaman, N. (2022). Towards industrial revolution 5.0 and explainable artificial intelligence: Challenges and opportunities. *International Journal of Computing and Digital Systems*, 12(1), 295-320.
- Ullah, I., Khan, I. U., Ouaisa, M., Ouaisa, M., & El Hajjami, S. (Eds.). (2024). *Future communication systems using artificial intelligence, internet of things and data science*. CRC Press.
- Van Hoang, T. (2024). Impact of integrated artificial intelligence and internet of things technologies on smart city transformation. *Journal of technical education science*, 19(Special Issue 01), 64-73.
- Zaidan, A. A., & Zaidan, B. B. (2020). A review on intelligent process for smart home applications based on IoT: coherent taxonomy, motivation, open challenges, and recommendations. *Artificial Intelligence Review*, 53(1), 141-165.
- Zhang, Z., Wen, F., Sun, Z., Guo, X., He, T., & Lee, C. (2022). Artificial intelligence-enabled sensing technologies in the 5G/internet of things era: from virtual reality/augmented reality to the digital twin. *Advanced Intelligent Systems*, 4(7), 2100228.

ALGORITMA CERDAS

Untuk Aplikasi AI, IoT dan Otomatisasi

Dr. Joseph Teguh Santoso, S.Kom, M.Kom.

BIODATA PENULIS



Dr. Joseph Teguh Santoso, M.Kom memiliki Jabatan Akademik Lektor Kepala dan praktisi industri yang berpengalaman. Saat ini menjabat sebagai Rektor Universitas Sains dan Teknologi Komputer (Universitas STEKOM), salah satu universitas terkemuka di Jawa Tengah, Indonesia. Dengan pengalaman lebih dari 13 tahun di dunia bisnis dan praktisi industri di China, beliau membawa perspektif global dan inovasi yang signifikan ke dalam dunia akademis. Sebagai seorang entrepreneur, penulis adalah pencipta TopLoker.com, sebuah platform inovatif yang merevolusi cara mencari dan menawarkan pekerjaan. TopLoker.com adalah portal lowongan bursa kerja terbesar di Indonesia, khusus untuk pendidikan SMA/SMK sederajat.

TopLoker.com telah mendapatkan penghargaan sebagai juara 1 Startup4Industry 2022 oleh Kementerian Perindustrian Republik Indonesia. Kontribusi Dr. Joseph dalam menyediakan akses pekerjaan yang luas bagi lulusan SMA/SMK telah membantu banyak individu menemukan peluang kerja yang sesuai dengan keahlian mereka. Selain itu, Dr. Joseph Teguh Santoso, M.Kom adalah pendiri dari dua organisasi yaitu (1) organisasi guru/pendidik PTIC (Perkumpulan Teacherpreneur Indonesia Cerdas) yang bertujuan untuk meningkatkan kualitas pendidikan dan kesejahteraan guru/pendidik dengan wawasan entrepreneurship, serta (2) organisasi industri PERKIVI (Perkumpulan Komunitas Industri dan Vokasi Indonesia) yang berfokus pada pengembangan link and match antara industri dan dunia pendidikan. Sebagai Rektor, Dr. Joseph Teguh Santoso, M.Kom memiliki kepemimpinan yang berorientasi pada hasil, dan berkomitmen untuk mendorong kemajuan Universitas Sains dan Teknologi Komputer (Universitas STEKOM). Saat ini Universitas STEKOM telah mengalami transformasi positif dalam peningkatan kualitas pendidikan, perluasan fasilitas, serta penguatan kemitraan Perguruan Tinggi Nasional dan Internasional. Beliau memprioritaskan pengembangan sumber daya manusia dan penelitian, serta memastikan bahwa universitas berada di garis depan dalam inovasi dan teknologi untuk mencapai tujuan akhir, yaitu lulusan yang mampu bekerja dan sukses setelah lulus. Dr. Joseph Teguh Santoso, M.Kom sering diundang sebagai pembicara di berbagai konferensi nasional maupun internasional dan telah menerima berbagai penghargaan atas dedikasinya dalam bidang pendidikan, industri, dan kewirausahaan.



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7695-41-3 (PDF)



9

786347

695413