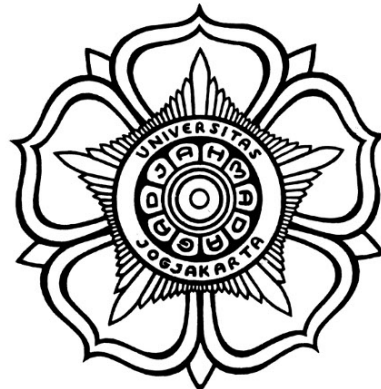


SKRIPSI

**PEMBELAJARAN REPRESENTASI SPASIAL-TEMPORAL BERBASIS
JARINGAN SARAF GRAF UNTUK OPTIMASI PEMBOBOTAN PADA
MODEL REGRESI DENGAN KOEFISIEN BERVARIASI DAN REGRESI
LOKAL TERBOBOTI**

***SPATIOTEMPORAL REPRESENTATION LEARNING BASED ON
GRAPH NEURAL NETWORKS FOR WEIGHT OPTIMIZATION IN
VARYING COEFFICIENT REGRESSION AND LOCALLY WEIGHTED
REGRESSION***



Kurniawan Chandra Wijaya
22/497908/PA/21466

**PROGRAM SARJANA PROGRAM STUDI STATISTIKA
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

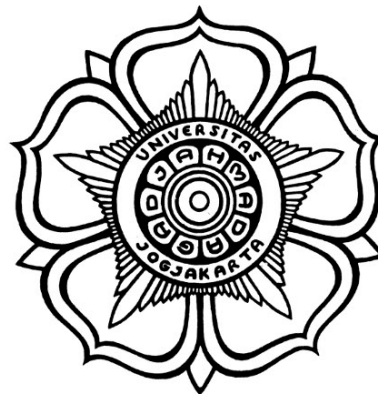
2025

SKRIPSI

**PEMBELAJARAN REPRESENTASI SPASIAL-TEMPORAL BERBASIS
JARINGAN SARAF GRAF UNTUK OPTIMASI PEMBOBOTAN PADA
MODEL REGRESI DENGAN KOEFISIEN BERVARIASI DAN REGRESI
LOKAL TERBOBOTI**

***SPATIOTEMPORAL REPRESENTATION LEARNING BASED ON
GRAPH NEURAL NETWORKS FOR WEIGHT OPTIMIZATION IN
VARYING COEFFICIENT REGRESSION AND LOCALLY WEIGHTED
REGRESSION***

Diajukan untuk memenuhi salah satu syarat memperoleh derajat
Sarjana Program Sarjana Program Studi Statistika



Kurniawan Chandra Wijaya
22/497908/PA/21466

**PROGRAM SARJANA PROGRAM STUDI STATISTIKA
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2025

HALAMAN PENGESAHAN

SKRIPSI

**PEMBELAJARAN REPRESENTASI SPASIAL-TEMPORAL BERBASIS
JARINGAN SARAF GRAF UNTUK OPTIMASI PEMBOBOTAN PADA
MODEL REGRESI DENGAN KOEFISIEN BERVARIASI DAN REGRESI
LOKAL TERBOBOTI**

Telah dipersiapkan dan disusun oleh

Kurniawan Chandra Wijaya
22/497908/PA/21466

Telah dipertahankan di depan Tim Penguji
pada tanggal tanggal bulan tahun

Susunan Tim Penguji

Prof. Dr. Abdurakhman, S.Si., M.Si.
Pembimbing Utama

Ketua Penguji
Ketua Tim Penguji

Penguji 1
Penguji

Penguji 2
Penguji

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Kurniawan Chandra Wijaya
NIM : 22/497908/PA/21466
Tahun terdaftar : 2025
Program Studi : Program Sarjana Program Studi Statistika
Fakultas : Fakultas Matematika dan Ilmu Pengetahuan Alam

menyatakan bahwa dalam dokumen ilmiah skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, tanggal bulan tahun

Materai Rp10.000,00

Kurniawan Chandra Wijaya
22/497908/PA/21466

Karya sederhana ini penulis persembahkan untuk ibu,
bapak, dan diri penulis.

Man jadda wajada.

Barang siapa bersungguh-sungguh, maka ia akan berhasil.

PRAKATA

Bismillahirrahmanirrahim. Puji syukur kepada Allah SWT atas limpahan berkah dan rahmat yang diberikan sehingga penulis dapat menyelesaikan tugas akhir yang berjudul ... sebagai salah satu syarat mendapatkan gelar sarjana di program studi statistika Universitas Gadjah Mada. Tugas akhir ini merupakan salah satu hasil dari proses belajar yang penulis jalani selama empat tahun berkuliah di universitas tercinta.

Penulis sadar bahwa dalam proses penyusunan tugas akhir ini tidak akan berjalan tanpa adanya dukungan dan bimbingan dari berbagai pihak. Maka dari itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Prof. Dr.Eng Kuwat Triyana, M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.
2. Bapak Dr. Nanang Susyanto, M.Sc., selaku Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.
3. Bapak Prof. Dr.rer.nat Dedi Rosadi, M.Sc., selaku Ketua Program Studi Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.

Seperti manusia pada umumnya, penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, semua kritik dan saran dari pembaca menjadi masukan yang sangat berharga bagi penulis. Semoga tugas akhir ini dapat bermanfaat untuk pembaca. Terima kasih.

Yogyakarta, Juni 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN BEBAS PLAGIASI	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
PRAKATA	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
INTISARI	xii
ABSTRACT	xiii
I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Pembatasan Masalah	3
1.3. Tujuan dan Manfaat Penelitian	3
1.4. Tinjauan Pustaka	4
1.5. Metodologi Penelitian	4
1.6. Sistematika Penulisan	5
II LANDASAN TEORI	6
2.1. Aljabar Matriks dan Tensor	6
2.1.1. Ruang Vektor dan Matriks	6
2.1.2. Sifat-Sifat Matriks	12
2.1.3. Invers Matriks	20
2.1.4. Determinan dan <i>Rank</i>	23
2.1.5. Nilai Eigen dan Vektor Eigen	29
2.1.6. Tensor	31
2.1.7. Dekomposisi Matriks dan Tensor	36
2.1.8. Diferensial Vektor dan Matriks	44
2.2. Regresi Spasial dan Spasial-Temporal	46
2.2.1. Regresi Linear Biasa atau <i>Ordinary Least Squares</i> (OLS)	46
2.2.2. Regresi Terboboti	52
2.2.3. Model Koefisien Bervariasi	56
2.2.4. Regresi dengan Dependensi Spasial dan Spasial Temporal	60

2.2.5.	Regresi dengan Heterogenitas Spasial dan Spasial Temporal	62
2.3.	Jaringan Saraf Tiruan	71
2.3.1.	Model Dasar dan Notasi	71
2.3.2.	Fungsi Aktivasi	71
2.3.3.	Fungsi Kerugian dan Kriteria Pembelajaran	75
2.3.4.	Pembelajaran dengan Propagasi Mundur	76
2.3.5.	Optimisasi Parameter	78
2.3.6.	Jaringan Saraf Tiruan sebagai Aproksimasi Universal	81
2.4.	Jaringan Saraf Graf	81
2.4.1.	Dasar Graf dan Laplacian	81
2.4.2.	Kerangka Penyampaian Pesan pada Jaringan Saraf	83
2.4.3.	Arsitektur Umum dalam GNN	84
2.4.4.	Kekuatan Representasi GNN	87
2.5.	Pembelajaran Representasi	88
2.5.1.	Landasan Teoretis Representasi	90
2.6.	Pembelajaran Semi-Terawasi	92
2.6.1.	Kerangka Dasar	92
2.6.2.	Asumsi Fundamental dalam Pembelajaran Semi-Terawasi	94
2.6.3.	Fungsi Kerugian untuk Pembelajaran Semi-Terawasi	95
III PENGEMBANGAN PEMBOBOTAN BERBASIS JARINGAN SARAF GRAF UNTUK REGRESI SPASIAL-TEMPORAL		100
3.1.	Kajian Literatur Terkait	100
3.1.1.	<i>Geographically Weighted Artificial Neural Network</i> (GWANN)	100
3.1.2.	<i>Spatial and Attribute Neural Network Weighted Regression</i> (SANNWR)	101
3.1.3.	<i>Spatial Regression Graph Convolutional Neural Networks</i> (SRGCNN)	101
3.1.4.	<i>Spatio-Temporal Graph Convolutional Networks</i> (ST-GCN)	102
3.1.5.	<i>Geographically Neural Network Weighted Regression</i> (GN-NWR)	102
3.2.	Kerangka Metodologis	105
3.2.1.	Kerangka Umum dan Alur Metodologi	105
3.2.2.	Alur Pembelajaran GNN-GTVC dan GNN-GTWR	107
3.2.3.	Perbandingan Konseptual Awal	108
3.3.	Regresi Spasial-Temporal Berkoeffisien Bervariasi berbasis GNN (GNN-GTVC)	109
3.3.1.	Formulasi Matematis	109

3.3.2. Pembentukan Bobot Koefisien Lokal melalui Representasi Graf	110
3.3.3. Estimasi Koefisien Global dan Prediksi	110
3.4. Regresi Spasial-Temporal Terboboti berbasis GNN (GNN-GTWR) .	111
3.4.1. Formulasi Estimator WLS pada Konteks Graf	111
3.4.2. Pembobotan Antartetangga dengan Representasi GNN . . .	112
3.5. Arsitektur Jaringan Graf untuk Pembelajaran Bobot	112
3.6. Pilihan Pemetaan Representasi ke Bobot	113
3.6.1. Fungsi Objektif dan Proses Pelatihan	115
3.7. Pembelajaran Semi-Terawasi (<i>Semi-Supervised Learning</i> , SSL) . . .	117
3.7.1. Kerangka Dasar dan Asumsi Fundamental	117
3.7.2. Fungsi Kerugian Semi-Terawasi	117
3.7.3. Integrasi SSL ke dalam Pelatihan GNN-GTVC dan GNN-GTWR	119
IV STUDI KASUS	121
4.1. Catatan Penting	121
4.2. Pembahasan	121
4.2.1. Pembahasan 1	121
4.2.2. Pembahasan 2	121
V PENUTUP	123
5.1. Kesimpulan	123
5.2. Saran	123
DAFTAR PUSTAKA	124
A Data	128
B Syntax R	129

DAFTAR TABEL

DAFTAR GAMBAR

2.1	Ilustrasi transformasi linear $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ dengan $f(x, y) = (2x, y)$. (Sumber: Dokumen penulis)	24
2.2	Ilustrasi bentuk bilinear anti-simetri $\omega(v_1, v_2) = x_1y_2 - x_2y_1$ pada \mathbb{R}^2 . (Sumber: Dokumen penulis)	26
2.3	Perbandingan luas parallelogram sebelum dan sesudah transformasi A . (Sumber: Dokumen penulis)	28
2.4	Diagram sifat universal. (Sumber: Dokumen penulis)	33
2.5	Ilustrasi (a) Dekomposisi Tucker dan (b) Dekomposisi CP pada Tensor. (Sumber: Jiang, Huang, Zhao, & Deng, 2020)	43
2.6	Ilustrasi dua graf isomorfik, yaitu \mathcal{G}_1 dan \mathcal{G}_2 memiliki struktur yang sama walaupun label simpul berbeda. (Sumber: Dokumen penulis)	87
3.1	Diagram Kerangka Kerja GTNNWR (Yin, Huang, Du, Wu, dkk., 2024)	104
3.2	Alur metodologi GNN-GTVC dan GNN-GTWR. (Sumber: Dokumen penulis)	106
4.1	SKRIPSI TU DIKERJAIN	121

INTISARI

PEMBELAJARAN REPRESENTASI SPASIAL-TEMPORAL BERBASIS JARINGAN SARAF GRAF UNTUK OPTIMASI PEMBOBOTAN PADA MODEL REGRESI DENGAN KOEFISIEN BERVARIASI DAN REGRESI LOKAL TERBOBOTI

Oleh

Kurniawan Chandra Wijaya

22/497908/PA/21466

TULIS ABSTRAK,

Kata Kunci: kata kunci.

ABSTRACT

SPATIOTEMPORAL REPRESENTATION LEARNING BASED ON GRAPH NEURAL NETWORKS FOR WEIGHT OPTIMIZATION IN VARYING COEFFICIENT REGRESSION AND LOCALLY WEIGHTED REGRESSION

By

Kurniawan Chandra Wijaya

22/497908/PA/21466

TULIS ABSTRAK ENGGRES,

Keyword: keywordsssss

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perekonomian Indonesia senantiasa menghadapi dinamika yang kompleks seiring dengan variasi spasial dan temporal yang terjadi di setiap wilayah. Fenomena inflasi, misalnya, tidak hanya dipengaruhi oleh faktor makroekonomi secara nasional, tetapi juga oleh karakteristik lokal seperti struktur industri, tingkat pendidikan, dan infrastruktur wilayah. Penelitian Amri (2020) menunjukkan bahwa model *Space-Time Autoregressive* (STAR) dan *Generalized STAR* (GSTAR) mampu menangkap keterkaitan spasial antarwilayah dalam peramalan indeks harga konsumen (IHK) di beberapa kota besar di Sumatra. Namun, kedua model tersebut masih bersifat linear dan global sehingga belum sepenuhnya mampu menggambarkan ketidakstasioneran hubungan ekonomi antarwilayah dan antarwaktu.

Dalam literatur ekonometrika spasial, ketidakstasioneran spasial (*spatial heterogeneity*) dan ketergantungan spasial (*spatial dependence*) menjadi dua aspek penting yang perlu diperhatikan (Anselin, 1988). Untuk mengakomodasi hal tersebut, Fotheringham, Brunson, and Charlton (2002) mengembangkan model *Geographically Weighted Regression* (GWR) yang memungkinkan koefisien regresi bervariasi antar lokasi. Model ini kemudian diperluas oleh Huang, Wu, and Barry (2010) menjadi *Geographically and Temporally Weighted Regression* (GTWR) dengan menambahkan dimensi waktu, sehingga mampu menangkap dinamika spasial sekaligus temporal dalam satu kerangka analisis. Walaupun interpretatif, model GTWR masih bergantung pada asumsi bentuk fungsi kernel dan pemilihan *bandwidth* yang bersifat subjektif serta tidak mampu merepresentasikan hubungan nonlinier yang kompleks antara jarak spasial dan bobot pembobotan.

Perkembangan terkini di bidang *geospatial artificial intelligence* (GeoAI) menghadirkan pendekatan *machine learning* yang dapat mempelajari pola nonline-

ar secara adaptif. Salah satunya adalah *Geographically Weighted Artificial Neural Network* (GWANN) yang dikembangkan oleh Hagenauer and Helbich (2022) serta *Spatial and Attribute Neural Network Weighted Regression* (SANNWR) oleh Ni, Wu, and Zhang (2022). Keduanya berupaya memanfaatkan jaringan saraf tiruan untuk mengestimasi fungsi pembobot spasial yang kompleks. Lebih lanjut, Yindkk. (2024) memperkenalkan model *Geographically Neural Network Weighted Regression* (GNNWR) dan *Geographically and Temporally Neural Network Weighted Regression* (GTNNWR), yang memadukan kerangka *varying coefficient* dengan kemampuan pembelajaran representasi nonlinear dari *neural networks*. Model tersebut menunjukkan peningkatan akurasi yang signifikan dalam memetakan hubungan spasial-temporal pada data lingkungan dan sosial.

Di sisi lain, Zhu, Liu, Yao, and Fischer (2022) mengusulkan *Spatial Regression Graph Convolutional Neural Networks* (SRGCNN) yang mengintegrasikan *graph convolutional neural network* (GCN) ke dalam paradigma regresi spasial. Pendekatan ini memperlakukan data spasial sebagai graf, dengan simpul merepresentasikan lokasi dan sisi (*edges*) merepresentasikan keterhubungan spasial antarunit observasi. Dengan mekanisme *message passing* yang mengagregasi informasi dari tetangga terdekat, SRGCNN tidak hanya mampu menangani struktur data non-Euclidean, tetapi juga mendukung pembelajaran semi-terawasi (*semi-supervised learning*) yang memungkinkan model belajar dari data yang sebagian tidak teramati. Pendekatan ini membuka peluang besar untuk memperkuat model regresi spasial konvensional melalui pembobotan berbasis graf.

Berdasarkan perkembangan tersebut, penelitian ini berupaya mengintegrasikan prinsip pembobotan adaptif berbasis GNN sebagaimana pada GNNWR dan SRGCNN ke dalam kerangka regresi spasial-temporal seperti GTWR. Model yang diusulkan adalah *Graph Neural Network–Geographically and Temporally Varying Coefficient* (GNN-GTVC) serta *Graph Neural Network–Geographically and Temporally Weighted Regression* (GNN-GTWR). Kedua model ini dirancang untuk menangkap dinamika spasial-temporal dengan lebih fleksibel melalui pembelajaran berbasis graf yang mengoptimasi matriks bobot. Dalam konteks empiris, model

ini akan diterapkan pada analisis spasial-temporal inflasi antarprovinsi di Indonesia periode 2024–2025, guna memahami disparitas inflasi antarwilayah dan dinamika faktor-faktor ekonomi yang memengaruhinya. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi metodologis dalam bidang ekonometrika spasial sekaligus implikasi kebijakan dalam pengendalian inflasi regional di Indonesia.

1.2. Pembatasan Masalah

Penelitian ini berfokus pada pengembangan dan penerapan model GNN-GTWR dan GNN-GTVC untuk menganalisis dinamika spasial-temporal inflasi antarprovinsi di Indonesia. Ruang lingkup penelitian dibatasi pada level provinsi sebagai unit analisis dengan periode pengamatan Januari 2024 hingga Agustus 2025. Variabel yang digunakan meliputi indikator makroekonomi utama seperti inflasi, upah minimum, tingkat pengangguran, dan indeks pembangunan manusia (IPM) yang bersumber dari Badan Pusat Statistik (BPS) dan Bank Indonesia (BI). Penelitian ini tidak mencakup analisis mikroekonomi individu maupun agregasi lintas negara, serta tidak membahas secara mendalam mekanisme kausalitas antarvariabel.

1.3. Tujuan dan Manfaat Penelitian

Tujuan utama penelitian ini adalah mengembangkan model pembelajaran spasial-temporal berbasis graf untuk menganalisis variasi inflasi antarwilayah di Indonesia. Secara khusus, penelitian ini bertujuan untuk:

1. mengonstruksi model *Graph Neural Network–Geographically and Temporally Weighted Regression* (GNN-GTWR) dan *Graph Neural Network–Geographically and Temporally Varying Coefficient* (GNN-GTVC) sebagai perluasan dari model GTWR dan VCM yang mampu menangkap ketidakstasioneran spasial dan temporal secara simultan melalui pembobotan adaptif berbasis GNN;
2. menerapkan model yang dikembangkan pada studi kasus inflasi antarpro-

vinsi di Indonesia periode 2024–2025 untuk mengidentifikasi pola spasial-temporal dan faktor-faktor ekonomi yang berpengaruh; dan

3. mengevaluasi kinerja model terhadap model konvensional (GTWR dan GWR) dalam hal ketepatan estimasi, stabilitas, dan interpretabilitas koefisien lokal.

Secara praktis, hasil penelitian ini diharapkan dapat memberikan kontribusi bagi lembaga perencana kebijakan, khususnya Bank Indonesia dan Kementerian Keuangan, dalam memahami dinamika inflasi regional dan merancang kebijakan pengendalian harga yang adaptif terhadap kondisi spasial-temporal.

1.4. Tinjauan Pustaka

Penelitian mengenai regresi spasial dan spasial-temporal telah berkembang pesat dalam dua dekade terakhir. Model klasik seperti GWR (Fotheringham dkk., 2002) dan GTWR (Huang dkk., 2010) menjadi dasar utama dalam analisis hubungan spasial yang tidak stasioner. Namun, keterbatasannya pada asumsi linearitas dan bentuk kernel mendorong munculnya model berbasis kecerdasan buatan seperti GWANN (Hagenauer & Helbich, 2022), SANNWR (Ni dkk., 2022), dan GNNWR (Yin dkk., 2024). Di sisi lain, pendekatan *graph-based learning* seperti SRGCNN (Zhu dkk., 2022) dan GSTRGCN (Xiong, Zhang, Wang, & Liu, 2024) memperlihatkan potensi besar dalam memodelkan struktur dependensi spasial-temporal yang kompleks melalui pembelajaran representasi graf. Berangkat dari temuan tersebut, penelitian ini berupaya menggabungkan kekuatan metodologis antara regresi terboti spasial-temporal dan pembelajaran berbasis graf guna mengembangkan model yang lebih adaptif, akurat, dan interpretatif.

1.5. Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi studi literatur dan studi kasus. Studi literatur dilakukan untuk mengkaji teori dan penelitian terdahulu terkait regresi spasial-temporal, jaringan saraf graf, serta pendekatan pembelajaran semi-terawasi yang relevan dengan pengembangan model GNN-GTWR dan GNN-GTVC. Studi kasus dilakukan dengan menggunakan data sekunder dari BPS

dan BI, yang mencakup 38 provinsi di Indonesia untuk periode Januari 2024 hingga Agustus 2025. Analisis dan komputasi dilakukan menggunakan bahasa pemrograman Python dengan pustaka *PyTorch Geometric* untuk implementasi jaringan saraf graf, serta perangkat lunak statistik seperti R atau GeoDa untuk validasi spasial.

1.6. Sistematika Penulisan

Pada penyusunan skripsi ini, penulis mengacu pada sistematika penulisan sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, pembatasan masalah, tujuan dan manfaat penelitian, tinjauan pustaka, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas dasar-dasar teori yang digunakan, termasuk teori regresi spasial-temporal, jaringan saraf graf, dan pembelajaran semi-terawasi.

BAB III PENGEMBANGAN PEMBOBOTAN BERBASIS JARINGAN SARAF GRAF UNTUK REGRESI SPASIAL-TEMPORAL

Bab ini menjelaskan rancangan model GNN-GTVC dan GNN-GTWR, formulasi matematis, serta prosedur estimasi dan validasi model.

BAB IV STUDI KASUS

Bab ini berisi hasil implementasi model pada data inflasi antarprovinsi di Indonesia, analisis spasial-temporal, serta perbandingan kinerja model.

BAB V PENUTUP

Bab ini menyajikan kesimpulan yang diperoleh dari hasil penelitian dan memberikan saran untuk penelitian selanjutnya.

BAB II

LANDASAN TEORI

Pada bab ini akan dibahas mengenai konsep, teori, dan referensi yang menjadi dasar dalam melakukan analisis suatu metode statistik. Konsep dasar yang dibahas pada bab ini adalah aljabar matriks dan tensor, regresi spasial dan spasial-temporal, jaringan saraf tiruan, jaringan saraf graf, pembelajaran representasi, dan pembelajaran semi-terawasi.

2.1. Aljabar Matriks dan Tensor

Aljabar linear merupakan salah satu fondasi matematis terpenting dalam statistika, ekonometrika, serta pembelajaran mesin. Banyak metode regresi spasial maupun spasial-temporal, termasuk model berbasis jaringan saraf graf atau *graph neural networks* (GNN), dapat diformulasikan dalam kerangka operasi matriks dan tensor. Oleh karena itu, pemahaman yang kuat mengenai struktur vektor, matriks, dan tensor serta sifat-sifat aljabarnya diperlukan sebelum membahas regresi spasial-temporal berbobot.

2.1.1. Ruang Vektor dan Matriks

Pembahasan aljabar linear umumnya berangkat dari konsep lapangan atau *field* sebagai struktur aljabar dasar tempat bilangan berlaku. Dari lapangan, dibangun ruang vektor, kemudian matriks sebagai representasi transformasi linear, dan tensor sebagai generalisasi multidimensi.

Definisi 2.1.1 (Lapangan, (Lang, 1987)) Misalkan K adalah subhimpunan dari bilangan kompleks \mathbb{C} , K disebut sebagai lapangan apabila memenuhi kondisi berikut.

- (a) Apabila x, y adalah elemen dari K , maka $x + y$ dan xy juga merupakan elemen dari K .

(b) Apabila $x \in K$, maka $-x$ juga merupakan elemen dari K . Lebih lanjut, jika $x \neq 0$, maka x^{-1} merupakan elemen dari K .

(c) 0 dan 1 merupakan elemen dari K .

Berdasarkan definisi di atas, dapat diperhatikan bahwa \mathbb{R} dan \mathbb{C} merupakan lapangan.

Contoh 2.1.2 Apabila dinotasikan \mathbb{Q} sebagai himpunan bilangan rasional, yaitu

$$\mathbb{Q} = \left\{ \frac{m}{n} : m, n \in \mathbb{Z}, n \neq 0 \right\}, \quad (2.1)$$

maka dapat dinyatakan bahwa \mathbb{Q} membentuk suatu lapangan. Misalkan $\frac{a}{b}, \frac{c}{d} \in \mathbb{Q}$ dengan $a, b, c, d \in \mathbb{Z}$ dan $b, d \neq 0$. Jelas bahwa

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd} \in \mathbb{Q}, \quad (2.2)$$

sehingga sifat tertutup terhadap penjumlahan, sebagai salah satu aksioma lapangan, terpenuhi. Lebih lanjut, untuk setiap $\frac{a}{b} \in \mathbb{Q}$ berlaku pula bahwa $-\frac{a}{b} \in \mathbb{Q}$. Selain itu, jika $a \neq 0$, maka invers perkalian $\frac{a}{b}$ diberikan oleh

$$\left(\frac{a}{b} \right)^{-1} = \frac{b}{a} \in \mathbb{Q}. \quad (2.3)$$

Dengan demikian, keberadaan invers aditif dan invers perkalian juga terjamin. Akhirnya, apabila dipilih $\frac{a}{b} = \frac{0}{1}$ dan $\frac{a}{b} = \frac{1}{1}$, diperoleh bahwa elemen identitas penjumlahan 0 dan identitas perkalian 1 termasuk dalam \mathbb{Q} . Di sisi lain, dapat dengan mudah diperiksa bahwa \mathbb{Z} bukan merupakan lapangan, karena apabila $n \in \mathbb{Z}$ dan $n \neq 0$, jelas bahwa $n^{-1} \notin \mathbb{Z}$, kecuali untuk kasus $n = 1$ atau $n = -1$.

Dalam ruang vektor dan matriks, dikenal juga istilah sublapangan. Apabila K dan L keduanya merupakan lapangan, serta dimisalkan bahwa $K \subseteq L$, maka K dapat disebut sebagai sublapangan atau *subfield* dari L . Elemen-elemen dari lapangan disebut sebagai bilangan atau skalar. (Lang, 1987)

Definisi 2.1.3 (Ruang Vektor, (Lang, 1987)) Suatu ruang vektor V atas suatu lapangan K adalah sebuah himpunan objek yang dapat dijumlahkan dan dikalikan dengan elemen-elemen dari K , sedemikian rupa sehingga hasil penjumlahan dua elemen dari V merupakan elemen V kembali, dan hasil perkalian sebuah elemen V dengan sebuah elemen dari K juga merupakan elemen dari V . Selain itu, sifat-sifat berikut dipenuhi:

- (a) $\forall u, v, w \in V$ berlaku $(u + v) + w = u + (v + w)$.
- (b) $\exists 0 \in V$ sedemikian sehingga $\forall u \in V$ berlaku $u + 0 = u$.
- (c) $\forall u \in V, \exists -u \in V$ sedemikian sehingga $u + (-u) = 0$.
- (d) $\forall u, v \in V$ berlaku $u + v = v + u$.
- (e) $\forall c \in K$ dan $\forall u, v \in V$ berlaku $c(u + v) = cu + cv$.
- (f) $\forall a, b \in K$ dan $\forall v \in V$ berlaku $(a + b)v = av + bv$.
- (g) $\forall a, b \in K$ dan $\forall v \in V$ berlaku $(ab)v = a(bv)$.
- (h) $\forall u \in V$ berlaku $1 \cdot u = u$, dengan 1 adalah elemen identitas pada K .

Dalam mendefinisikan ruang vektor, lapangan tempat ruang vektor berada harus didefinisikan dengan spesifik, misalnya \mathbb{C}^n merupakan ruang vektor atas \mathbb{C} , tetapi \mathbb{R}^n bukan ruang vektor atas \mathbb{C} . \mathbb{R}^n merupakan ruang vektor atas \mathbb{R} .

Definisi 2.1.4 (Ruang Matriks, (Lang, 1987)) Misalkan K merupakan lapangan dan $m, n \geq 1 \in \mathbb{Z}$. Matriks adalah suatu larik atau array dari bilangan-bilangan dalam K yang dinotasikan sebagai

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}.$$

Notasi matriks pada definisi di atas dapat dipersingkat dengan menulis (a_{ij}) , $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$. Matriks tersebut merupakan matriks $m \times n$ yang berarti bahwa matriks tersebut memiliki m -baris dan n -kolom. Sebagai contoh, kolom pertama dari matriks tersebut adalah

$$\begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix}$$

dan baris pertamanya adalah $(a_{11}, a_{12}, \dots, a_{1n})$. Nilai a_{ij} disebut sebagai komponen dari matriks.

Dua buah matriks dikatakan sama apabila keduanya memiliki ukuran yang sama dan elemen-elemen pada posisi yang bersesuaian juga sama. Oleh karena itu, jika $\mathbf{A} = (a_{ij})$ dan $\mathbf{B} = (b_{ij})$, maka $\mathbf{A} = \mathbf{B}$ apabila $a_{ij} = b_{ij}$, untuk setiap i, j .

Definisi 2.1.5 (Kebebasan Linear, (Lang, 1987)) Misalkan V ruang vektor atas lapangan K dan misalkan pula $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ elemen-elemen dari V , maka $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ dikatakan bebas linear jika dan hanya jika kapan pun a_1, a_2, \dots, a_n merupakan bilangan sehingga

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_n\mathbf{v}_n = \mathbf{0}, \quad (2.4)$$

maka $a_i = 0, \forall i = 1, 2, \dots, n$.

Vektor $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ dikatakan bergantung linear terhadap K apabila terdapat elemen-elemen a_1, a_2, \dots, a_n di K yang semuanya tidak sama dengan nol, sedemikian sehingga Persamaan (2.4) terpenuhi.

Contoh 2.1.6 Misalkan $V = K^n$ dan vektor $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ didefinisikan sebagai

berikut.

$$\begin{aligned}\mathbf{v}_1 &= (1, 0, \dots, 0) \\ &\vdots \\ \mathbf{v}_n &= (0, 0, \dots, 1).\end{aligned}$$

Vektor-vektor $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ dikatakan bebas linear. Misalkan a_1, a_2, \dots, a_n merupakan bilangan-bilangan sedemikian sehingga Persamaan (2.4) terpenuhi. Sebab

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_n\mathbf{v}_n = (a_1, a_2, \dots, a_n), \quad (2.5)$$

dapat disimpulkan bahwa $a_i = 0, \forall i = 1, 2, \dots, n$.

Definisi 2.1.7 (*Span* atau *Jangkauan Linear*, (Axler, 2024)) Misalkan V ruang vektor atas lapangan K dan $S = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\} \subseteq V$. *Span* dari S , ditulis $\text{span}(S)$, adalah himpunan semua kombinasi linear dari elemen-elemen S , yaitu

$$\text{span}(S) = \left\{ \sum_{i=1}^m a_i \mathbf{v}_i : a_i \in K \right\}. \quad (2.6)$$

Dengan kata lain, $\text{span}(S)$ adalah subruang terkecil dari V yang memuat S . Di sisi lain, span dari himpunan kosong $\{\}$ didefinisikan sebagai $\{0\}$.

Contoh 2.1.8 Pada \mathbb{R}^2 , ambil $S = \{(1, 0), (0, 1)\}$, maka

$$\text{span}(S) = \{a(1, 0) + b(0, 1) : a, b \in \mathbb{R}\} = \mathbb{R}^2. \quad (2.7)$$

Hal ini berarti dua vektor standar membentang atau menjangkau (*spanning*) seluruh bidang \mathbb{R}^2 . Pada \mathbb{R}^3 , ambil $S = \{(1, 0, 0), (0, 1, 0)\}$, maka

$$\text{span}(S) = \{a(1, 0, 0) + b(0, 1, 0) : a, b \in \mathbb{R}\} = \{(x, y, 0) : x, y \in \mathbb{R}\}, \quad (2.8)$$

yaitu bidang xy dalam \mathbb{R}^3 .

Definisi 2.1.9 (Basis, (Axler, 2024)) Suatu basis V adalah himpunan vektor-vektor di V yang bebas linear dan menjangkau (spanning) V .

Contoh 2.1.10 Perhatikan vektor-vektor $\mathbf{v}_1 = (1, 2, -4)$ dan $\mathbf{v}_2 = (7, -5, 6)$ di dalam \mathbb{R}^3 . Jelas bahwa \mathbf{v}_1 dan \mathbf{v}_2 bebas linear, sebab tidak terdapat skalar $c \in \mathbb{R}$ sehingga $\mathbf{v}_2 = c\mathbf{v}_1$. Namun, himpunan $\{\mathbf{v}_1, \mathbf{v}_2\}$ hanya terdiri dari dua vektor, sehingga $\text{span}\{\mathbf{v}_1, \mathbf{v}_2\}$ paling jauh merupakan subruang berdimensi 2, yaitu sebuah bidang melalui titik asal di \mathbb{R}^3 . Sebab $\text{span}\{\mathbf{v}_1, \mathbf{v}_2\} \subsetneq \mathbb{R}^3$, maka himpunan ini tidak dapat dijadikan basis dari \mathbb{R}^3 .

Teorema 2.1.11 (Karakterisasi Basis, (Axler, 2024)) Sebuah list dari vektor-vektor $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ merupakan basis dari V jika dan hanya jika $\forall \mathbf{v} \in V$ dapat dituliskan secara unik dalam bentuk

$$\mathbf{v} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_n\mathbf{v}_n, \quad (2.9)$$

dengan $a_1, a_2, \dots, a_n \in \mathbb{R} \cup \mathbb{C}$.

Bukti. Pertama andaikan bahwa $\mathbf{v}_1, \dots, \mathbf{v}_n$ adalah basis dari V . Sebab $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ menjangkau V , maka untuk setiap $\mathbf{v} \in V$ terdapat skalar $a_1, \dots, a_n \in F$ sehingga Persamaan (2.9) terpenuhi. Untuk menunjukkan bahwa representasi tersebut unik, misalkan terdapat skalar $c_1, \dots, c_n \in F$ dengan

$$\mathbf{v} = c_1\mathbf{v}_1 + \dots + c_n\mathbf{v}_n. \quad (2.10)$$

Dengan mengurangkan Persamaan (2.10) dari Persamaan (2.9) akan diperoleh

$$0 = (a_1 - c_1)\mathbf{v}_1 + \dots + (a_n - c_n)\mathbf{v}_n. \quad (2.11)$$

Sebab $\mathbf{v}_1, \dots, \mathbf{v}_n$ bebas linear, maka setiap koefisien harus nol, yaitu $a_k - c_k = 0$ untuk $k = 1, \dots, n$. Dengan demikian $a_1 = c_1, \dots, a_n = c_n$, sehingga representasi (2.9) bersifat unik. Hal ini menyelesaikan pembuktian arah pertama.

Sebaliknya, andaikan setiap $\mathbf{v} \in V$ dapat dituliskan secara unik dalam ben-

tuk

$$\mathbf{v} = a_1 \mathbf{v}_1 + \cdots + a_n \mathbf{v}_n, \quad (2.12)$$

maka jelas $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ menjangkau V . Untuk membuktikan bahwa $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ bebas linear, andaikan terdapat $a_1, \dots, a_n \in F$ sehingga

$$0 = a_1 \mathbf{v}_1 + \cdots + a_n \mathbf{v}_n. \quad (2.13)$$

Sebab representasi (2.12) bersifat unik, khususnya untuk $\mathbf{v} = 0$, maka harus berlaku $a_1 = \cdots = a_n = 0$. Jadi $\mathbf{v}_1, \dots, \mathbf{v}_n$ bebas linear. Dengan demikian $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ adalah basis V . ■

Definisi 2.1.12 (Ruang Vektor Berdimensi-Hingga (Axler, 2024)) Suatu ruang vektor V atas medan F dikatakan berdimensi-hingga apabila terdapat suatu basis $\{v_1, \dots, v_n\}$ dari V yang terdiri atas sejumlah hingga (n buah) vektor. Dengan kata lain, V berdimensi-hingga jika V dapat dijangkau oleh suatu himpunan vektor berukuran hingga.

Definisi 2.1.13 (Dimensi (Axler, 2024)) Dimensi dari suatu ruang vektor berdimensi-hingga V adalah banyaknya elemen dalam setiap basis V . Sebab semua basis dari V memiliki panjang yang sama, maka bilangan ini terdefinisi dengan baik. Dimensi V dilambangkan dengan $\dim V$.

Contoh 2.1.14 Jika $U = \{(x, x, y) \in \mathbb{R}^3 : x, y \in \mathbb{R}\}$, maka $\dim U = 2$ karena $(1, 1, 0), (0, 0, 1)$ merupakan basis dari U .

2.1.2. Sifat-Sifat Matriks

Seluruh bagian ini bekerja di atas suatu lapangan K (umumnya $K = \mathbb{R}$ atau $K = \mathbb{C}$). Untuk $m, n \geq 1$, himpunan seluruh matriks $m \times n$ berelemen di K dilambangkan dengan $\text{Mat}_{m \times n}(K)$. Dengan penjumlahan dan perkalian skalar yang didefinisikan komponen-demi-komponen, $\text{Mat}_{m \times n}(K)$ membentuk ruang vektor atas lapangan K (Axler, 2024).

Definisi 2.1.15 (Kesamaan, Penjumlahan, dan Perkalian dengan Skalar) Untuk

$\mathbf{A} = (a_{ij}), \mathbf{B} = (b_{ij}) \in \text{Mat}_{m \times n}(K)$ dan $c \in K$, berlaku beberapa hal berikut.

- (a) $\mathbf{A} = \mathbf{B}$ jika dan hanya jika $a_{ij} = b_{ij}$ untuk semua i, j ;
- (b) $(\mathbf{A} + \mathbf{B})_{ij} = a_{ij} + b_{ij}$ (didefinisikan hanya bila ukuran sama); dan
- (c) $(c\mathbf{A})_{ij} = c a_{ij}$.

Definisi 2.1.16 (Perkalian Matriks, (Axler, 2024)) Jika $\mathbf{A} \in \text{Mat}_{m \times n}(K)$ dan $\mathbf{B} \in \text{Mat}_{n \times p}(K)$, maka hasil kali $\mathbf{AB} \in \text{Mat}_{m \times p}(K)$ didefinisikan oleh

$$(\mathbf{AB})_{ik} = \sum_{j=1}^n a_{ij} b_{jk} \quad (1 \leq i \leq m, 1 \leq k \leq p). \quad (2.14)$$

Contoh 2.1.17 Perkalian matriks berukuran 3×2 dan 2×4 :

$$\underbrace{\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}}_{3 \times 2} \underbrace{\begin{pmatrix} 6 & 5 & 4 & 3 \\ 2 & 1 & 0 & -1 \end{pmatrix}}_{2 \times 4} = \underbrace{\begin{pmatrix} 10 & 7 & 4 & 1 \\ 26 & 19 & 12 & 5 \\ 42 & 31 & 20 & 9 \end{pmatrix}}_{3 \times 4}. \quad (2.15)$$

Proposisi 2.1.18 (Sifat-Sifat Elementer Perkalian Matriks, (Axler, 2024)) Untuk ukuran yang sesuai, berlaku:

- (a) sifat asosiatif, yaitu $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$;
- (b) sifat distributif, yaitu $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$ dan $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$;
- (c) sifat identitas, yaitu $\mathbf{I}_m \mathbf{A} = \mathbf{A}$ dan $\mathbf{A} \mathbf{I}_n = \mathbf{A}$ untuk $\mathbf{A} \in \text{Mat}_{m \times n}(K)$; dan
- (d) dapat terjadi $\mathbf{AB} \neq \mathbf{BA}$.

Bukti. Di bawah ini adalah pembuktian untuk keempat sifat elementer perkalian matriks.

(a) Untuk $(\mathbf{AB})\mathbf{C}$, berlaku

$$\begin{aligned}
 ((\mathbf{AB})\mathbf{C})_{ik} &= \sum_{\ell=1}^p (\mathbf{AB})_{i\ell} c_{\ell k} \\
 &= \sum_{\ell=1}^p \left(\sum_{j=1}^n a_{ij} b_{j\ell} \right) c_{\ell k} \\
 &= \sum_{j=1}^n \sum_{\ell=1}^p a_{ij} b_{j\ell} c_{\ell k}.
 \end{aligned} \tag{2.16}$$

Di sisi lain, untuk $\mathbf{A}(\mathbf{BC})$ berlaku

$$\begin{aligned}
 (\mathbf{A}(\mathbf{BC}))_{ik} &= \sum_{j=1}^n a_{ij} (\mathbf{BC})_{jk} \\
 &= \sum_{j=1}^n a_{ij} \left(\sum_{\ell=1}^p b_{j\ell} c_{\ell k} \right) \\
 &= \sum_{j=1}^n \sum_{\ell=1}^p a_{ij} b_{j\ell} c_{\ell k}.
 \end{aligned} \tag{2.17}$$

Keduanya merupakan matriks yang sama.

(b) Untuk $\mathbf{A}(\mathbf{B} + \mathbf{C})$, berlaku

$$\begin{aligned}
 (\mathbf{A}(\mathbf{B} + \mathbf{C}))_{ik} &= \sum_{j=1}^n a_{ij} (b_{jk} + c_{jk}) \\
 &= \sum_{j=1}^n a_{ij} b_{jk} + \sum_{j=1}^n a_{ij} c_{jk} \\
 &= (\mathbf{AB})_{ik} + (\mathbf{AC})_{ik}.
 \end{aligned} \tag{2.18}$$

Kasus $(\mathbf{A} + \mathbf{B})\mathbf{C}$ serupa.

(c) Untuk \mathbf{I}_m merupakan matriks identitas berukuran $m \times m$, berlaku

$$(\mathbf{I}_m \mathbf{A})_{ik} = \sum_{j=1}^m (\mathbf{I}_m)_{ij} a_{jk} = \sum_{j=1}^m \delta_{ij} a_{jk} = a_{ik}, \tag{2.19}$$

dan

$$(\mathbf{A}\mathbf{I}_n)_{ik} = \sum_{j=1}^n a_{ij}(\mathbf{I}_n)_{jk} = \sum_{j=1}^n a_{ij}\delta_{jk} = a_{ik}, \quad (2.20)$$

dengan

$$\delta_{ij} = \begin{cases} 1, & \text{jika } i = j, \\ 0, & \text{jika } i \neq j. \end{cases} \quad (2.21)$$

Dapat disimpulkan bahwa $\mathbf{IA} = \mathbf{A}$.

- (d) Sifat ini akan dibuktikan dengan *counterexample*. Pertimbangkan perkalian matriks berikut.

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \quad (2.22)$$

Apabila urutan dibalik, akan memberikan matriks $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, sehingga kontradiksi apabila $\mathbf{AB} = \mathbf{BA}$ untuk semua \mathbf{A} dan \mathbf{B} .

■

Definisi 2.1.19 (Perkalian Hadamard, (Horn & Johnson, 2012)) Jika diberikan $\mathbf{A} = (a_{ij}) \in \text{Mat}_{m \times n}$ dan $\mathbf{B} = (b_{ij}) \in \text{Mat}_{m \times n}$, maka perkalian Hadamard atau perkalian Schur dari \mathbf{A} dan \mathbf{B} merupakan perkalian elemen per elemen dari matriks tersebut atau dapat dituliskan sebagai berikut.

$$\mathbf{A} \odot \mathbf{B} = (a_{ij}b_{ij}) \in \text{Mat}_{m \times n}. \quad (2.23)$$

Notasi perkalian Hadamard \odot dalam beberapa literatur dapat dituliskan juga dengan \circ .

Contoh 2.1.20 Misalkan

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}.$$

Maka hasil perkalian Hadamard adalah

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} 1 \cdot 7 & 2 \cdot 8 & 3 \cdot 9 \\ 4 \cdot 10 & 5 \cdot 11 & 6 \cdot 12 \end{bmatrix} = \begin{bmatrix} 7 & 16 & 27 \\ 40 & 55 & 72 \end{bmatrix}.$$

Definisi 2.1.21 (*Transpose*) Jika $\mathbf{A} \in \text{Mat}_{m \times n}(K)$, maka transpose dari \mathbf{A} , dilambangkan sebagai \mathbf{A}^\top , dengan $\mathbf{A}^\top \in \text{Mat}_{n \times m}(K)$ didefinisikan oleh $(\mathbf{A}^\top)_{ji} = a_{ij}$.

Contoh 2.1.22 Salah satu contoh *transpose* dari matriks adalah

$$\mathbf{A} = \begin{pmatrix} -5 & 2 & 4 \\ 3 & 6 & -2 \end{pmatrix} \iff \mathbf{A}^\top = \begin{pmatrix} -5 & 3 \\ 2 & 6 \\ 4 & -2 \end{pmatrix}, \quad (2.24)$$

sedangkan untuk vektor adalah

$$\mathbf{a} = \begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix} \iff \mathbf{a}^\top = (2, -3, 1). \quad (2.25)$$

Notasi untuk *transpose* pada vektor \mathbf{a} dapat juga dinotasikan dengan \mathbf{a}' . Di sisi lain, untuk skalar di lapangan K atau $c \in K$, $c^\top = c$.

Teorema 2.1.23 (*Sifat-Sifat Transpose*) Untuk \mathbf{A}, \mathbf{B} berukuran sesuai dan $c \in K$, berlaku:

- (a) $(\mathbf{A}^\top)^\top = \mathbf{A}$;
- (b) $(\mathbf{A} + \mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top$;
- (c) $(c\mathbf{A})^\top = c\mathbf{A}^\top$; dan
- (d) $(\mathbf{AB})^\top = \mathbf{B}^\top \mathbf{A}^\top$.

Bukti. Berikut ini adalah bukti dari keempat sifat *transpose* tersebut.

(a) Elemen-elemen dari $(\mathbf{A}^\top)^\top$ akan sama dengan elemen-elemen dari \mathbf{A} , karena

$$((\mathbf{A}^\top)^\top)_{ij} = a_{ij} = (\mathbf{A})_{ij}. \quad (2.26)$$

(b) Elemen-elemen dari $(\mathbf{A} + \mathbf{B})^\top$ adalah $((\mathbf{A} + \mathbf{B})^\top)_{ij} = a_{ji} + b_{ji}$ yang merupakan elemen-elemen dari $\mathbf{A}^\top + \mathbf{B}^\top$.

(c) Elemen-elemen dari $(c\mathbf{A})^\top$ adalah ca_{ji} yang merupakan elemen-elemen dari $c\mathbf{A}^\top$.

(d) Dengan menuliskan $c_{ik} = (\mathbf{AB})_{ik} = \sum_j a_{ij}b_{jk}$ akan didapatkan

$$\begin{aligned} ((\mathbf{AB})^\top)_{ki} &= c_{ik} \\ &= \sum_j a_{ij}b_{jk} \\ &= \sum_j (\mathbf{B}^\top)_{kj}(\mathbf{A}^\top)_{ji} \\ &= (\mathbf{B}^\top \mathbf{A}^\top)_{ki}. \end{aligned} \quad (2.27)$$

■

Definisi 2.1.24 (Matriks Simetris, Diagonal, dan Identitas) Berikut ini adalah beberapa definisi terkait matriks simetris, diagonal, dan identitas.

(a) $\mathbf{A} \in \text{Mat}_{n \times n}(K)$ disebut simetris jika $\mathbf{A}^\top = \mathbf{A}$.

(b) \mathbf{A} disebut diagonal jika $a_{ij} = 0$ untuk $i \neq j$.

(c) \mathbf{I}_n adalah matriks identitas berukuran n , dengan elemen diagonal 1 dan selainnya 0.

Contoh 2.1.25 Matriks $\mathbf{A} = \begin{pmatrix} 3 & -2 & 4 \\ -2 & 10 & -7 \\ 4 & -7 & 9 \end{pmatrix}$ adalah matriks simetris karena $\mathbf{A}^\top = \mathbf{A}$.

Definisi 2.1.26 (Matriks Idempoten, (Dhrymes, 2013)) Suatu matriks persegi \mathbf{A} dikatakan idempoten jika dan hanya jika

$$\mathbf{A}\mathbf{A} = \mathbf{A}^2 = \mathbf{A}. \quad (2.28)$$

Definisi 2.1.27 (Jejak atau Trace) Untuk $\mathbf{A} = (a_{ij}) \in \text{Mat}_{n \times n}(K)$, jejak atau trace didefinisikan sebagai

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^n a_{ii}. \quad (2.29)$$

Teorema 2.1.28 (Sifat-Sifat Trace, (Axler, 2024)) Untuk ukuran yang sesuai, berlaku:

- (a) $\text{tr}(\alpha\mathbf{A} + \beta\mathbf{B}) = \alpha \text{tr}(\mathbf{A}) + \beta \text{tr}(\mathbf{B})$;
- (b) $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$, lebih umum $\text{tr}(\mathbf{ABC}) = \text{tr}(\mathbf{BCA})$; dan
- (c) jika \mathbf{B} invertibel, maka $\text{tr}(\mathbf{B}^{-1}\mathbf{AB}) = \text{tr}(\mathbf{A})$.

Bukti. Berikut ini adalah bukti-bukti untuk ketiga sifat di atas.

(a)

$$\begin{aligned} \text{tr}(\alpha\mathbf{A} + \beta\mathbf{B}) &= \sum_i (\alpha a_{ii} + \beta b_{ii}) \\ &= \alpha \sum_i a_{ii} + \beta \sum_i b_{ii} \\ &= \alpha \text{tr}(\mathbf{A}) + \beta \text{tr}(\mathbf{B}). \end{aligned} \quad (2.30)$$

(b) Untuk dua faktor,

$$\begin{aligned}
 \text{tr}(\mathbf{AB}) &= \sum_i (\mathbf{AB})_{ii} \\
 &= \sum_i \sum_j a_{ij} b_{ji} \\
 &= \sum_j \sum_i b_{ji} a_{ij} \\
 &= \sum_j (\mathbf{BA})_{jj} \\
 &= \text{tr}(\mathbf{BA}).
 \end{aligned} \tag{2.31}$$

Untuk tiga faktor,

$$\begin{aligned}
 \text{tr}(\mathbf{ABC}) &= \sum_i (\mathbf{ABC})_{ii} \\
 &= \sum_i \sum_j \sum_k a_{ij} b_{jk} c_{ki} \\
 &= \sum_j \sum_k \sum_i b_{jk} c_{ki} a_{ij} \\
 &= \text{tr}(\mathbf{BCA}).
 \end{aligned} \tag{2.32}$$

(c) Dengan sifat (b), dapat diketahui bahwa

$$\begin{aligned}
 \text{tr}(\mathbf{B}^{-1}\mathbf{AB}) &= \text{tr}(\mathbf{ABB}^{-1}) \\
 &= \text{tr}(\mathbf{AI}) \\
 &= \text{tr}(\mathbf{A}).
 \end{aligned} \tag{2.33}$$

■

Definisi 2.1.29 (Matriks Semi-Definit Positif) Misalkan \mathbf{A} merupakan suatu matriks persegi berdimensi m dan \mathbf{x} merupakan vektor dengan m elemen. \mathbf{A} dikatakan

semi-definit positif jika dan hanya jika untuk semua vektor \mathbf{x} berlaku

$$\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0. \quad (2.34)$$

Matriks \mathbf{A} dikatakan definit positif jika dan hanya jika untuk \mathbf{x} tak kosong berlaku

$$\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0. \quad (2.35)$$

Definisi 2.1.30 (Matriks Definit Negatif) *Misalkan matriks \mathbf{A} merupakan matriks persegi berukuran n , maka \mathbf{A} dikatakan definit negatif jika dan hanya jika $-\mathbf{A}$ definit positif. Hal yang sama juga berlaku untuk semi-definit negatif.*

Beberapa literatur menggunakan notasi \succeq untuk menunjukkan matriks semidefinit positif, seperti $\mathbf{A} \succeq 0$ berarti $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0, \forall \mathbf{x}$.

2.1.3. Invers Matriks

Sesuai kerangka Axler, konsep invers dipahami melalui operator linear dan representasinya dalam suatu basis. Sebuah matriks bujur sangkar merepresentasikan operator linear $T : K^n \rightarrow K^n$, dan invers matriks berkorespondensi dengan invers operator linear T^{-1} ketika T bijektif.

Definisi 2.1.31 (Invers Matriks, (Axler, 2024)) *Misalkan $\mathbf{A} \in K^{n \times n}$. Matriks \mathbf{A} disebut invertibel (atau non-singular) apabila terdapat matriks $\mathbf{B} \in K^{n \times n}$ sedemikian sehingga*

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n, \quad (2.36)$$

dengan \mathbf{I}_n matriks identitas berukuran $n \times n$. Matriks \mathbf{B} disebut invers dari \mathbf{A} dan dinotasikan \mathbf{A}^{-1} .

Teorema 2.1.32 (Keunikan Invers) *Jika \mathbf{A} invertibel, maka \mathbf{A}^{-1} tunggal.*

Bukti. Jika \mathbf{B} dan \mathbf{C} keduanya memenuhi $\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$ dan $\mathbf{AC} = \mathbf{CA} = \mathbf{I}_n$, maka

$$\mathbf{B} = \mathbf{BI}_n = \mathbf{B}(\mathbf{AC}) = (\mathbf{BA})\mathbf{C} = \mathbf{I}_n \mathbf{C} = \mathbf{C}. \quad (2.37)$$



Proposisi 2.1.33 (Sifat-Sifat Invers, (Axler, 2024)) Misalkan $\mathbf{A}, \mathbf{B} \in K^{n \times n}$ invertibel dan $\alpha \in K$. Berlaku:

- (a) $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$;
- (b) $(\mathbf{A}^\top)^{-1} = (\mathbf{A}^{-1})^\top$; pada $K = \mathbb{C}$ juga berlaku $(\mathbf{A}^*)^{-1} = (\mathbf{A}^{-1})^*$;
- (c) $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$;
- (d) Jika $\mathbf{A} = \text{diag}(\mathbf{B}_1, \dots, \mathbf{B}_k)$ dengan setiap \mathbf{B}_i invertibel (ukuran mungkin berbeda), maka

$$\mathbf{A}^{-1} = \text{diag}(\mathbf{B}_1^{-1}, \dots, \mathbf{B}_k^{-1}).$$

Salah satu cara atau algoritma perhitungan invers matriks adalah dengan invers Gauss-Jordan. Jika \mathbf{A} invertibel, maka terdapat barisan operasi baris elementer (matriks elementer $\mathbf{E}_k \cdots \mathbf{E}_1$) sehingga $\mathbf{E}_k \cdots \mathbf{E}_1 \mathbf{A} = \mathbf{I}$. Dengan demikian,

$$\mathbf{A}^{-1} = \mathbf{E}_k \cdots \mathbf{E}_1. \quad (2.38)$$

Secara praktis, dapat dilakukan lakukan reduksi baris teraugmentasi:

$$[\mathbf{A} \mid \mathbf{I}] \xrightarrow{\text{operasi baris}} [\mathbf{I} \mid \mathbf{A}^{-1}].$$

Contoh 2.1.34 Ambil

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}.$$

Dengan melakukan eliminasi Gauss–Jordan pada $[\mathbf{A} \mid \mathbf{I}]$ diperoleh

$$\left[\begin{array}{ccc|ccc} 2 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 0 & 1 \end{array} \right] \xrightarrow{R_1 \leftarrow R_1 - \frac{1}{2}R_2} \left[\begin{array}{ccc|ccc} \frac{3}{2} & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 0 & 1 \end{array} \right]$$

$$R_2 \leftarrow R_2 - R_3 \xrightarrow{\quad} \left[\begin{array}{ccc|ccc} \frac{3}{2} & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 1 & 1 & -1 & 0 & 1 & -1 \\ 0 & 1 & 2 & 0 & 0 & 1 \end{array} \right] \xrightarrow{R_3 \leftarrow R_3 - R_2} \left[\begin{array}{ccc|ccc} \frac{3}{2} & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 1 & 1 & -1 & 0 & 1 & -1 \\ -1 & 0 & 3 & 0 & -1 & 2 \end{array} \right]$$

$$R_1 \leftarrow R_1 + \frac{1}{2}R_3 \xrightarrow{\quad} \left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 0 & 1 & -1 \\ -1 & 0 & 3 & 0 & -1 & 2 \end{array} \right] \xrightarrow{R_2 \leftarrow R_2 - R_1} \left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & -1 & 1 \\ 0 & 1 & -2 & -1 & 2 & -2 \\ -1 & 0 & 3 & 0 & -1 & 2 \end{array} \right]$$

$$R_3 \leftarrow R_3 + R_1 \xrightarrow{\quad} \left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & -1 & 1 \\ 0 & 1 & -2 & -1 & 2 & -2 \\ 0 & 0 & 4 & 1 & -2 & 3 \end{array} \right] \xrightarrow{R_3 \leftarrow \frac{1}{4}R_3} \left[\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & -1 & 1 \\ 0 & 1 & -2 & -1 & 2 & -2 \\ 0 & 0 & 1 & \frac{1}{4} & -\frac{1}{2} & \frac{3}{4} \end{array} \right]$$

$$R_1 \leftarrow R_1 - R_3, R_2 \leftarrow R_2 + 2R_3 \xrightarrow{\quad} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{3}{4} & -\frac{1}{2} & \frac{1}{4} \\ 0 & 1 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{4} & -\frac{1}{2} & \frac{3}{4} \end{array} \right].$$

Maka

$$\mathbf{A}^{-1} = \begin{pmatrix} \frac{3}{4} & -\frac{1}{2} & \frac{1}{4} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2} & \frac{3}{4} \end{pmatrix}.$$

Contoh 2.1.35 (Invers 2×2 dengan Rumus Klasik) Untuk $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ dengan

$ad - bc \neq 0$,

$$\mathbf{A}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}. \quad (2.39)$$

Rumus ini konsisten dengan Gauss–Jordan dan dapat dipandang sebagai kasus khusus *adjoint* ketika $n = 2$.

2.1.4. Determinan dan Rank

Pada seluruh bagian ini, V menyatakan ruang vektor berdimensi hingga atas suatu lapangan K dan $\mathcal{L}(V, W)$ himpunan semua pemetaan linear $V \rightarrow W$.

Definisi 2.1.36 (Pemetaan Linear, (Axler, 2024)) Misalkan V, W ruang vektor atas suatu lapangan K . Suatu pemetaan $T : V \rightarrow W$ disebut pemetaan linear jika untuk semua $\mathbf{u}, \mathbf{v} \in V$ dan $\lambda \in K$ berlaku

$$T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v}), \quad T(\lambda \mathbf{v}) = \lambda T(\mathbf{v}). \quad (2.40)$$

Notasi $\mathcal{L}(V, W)$ menyatakan himpunan semua pemetaan linear dari V ke W . Jika $V = W$, maka $\mathcal{L}(V) = \mathcal{L}(V, V)$ adalah himpunan semua operator linear pada V .

Contoh 2.1.37 Berikut ini adalah beberapa contoh pemetaan linear.

- (a) Jika $V = W = \mathbb{R}^2$, maka pemetaan

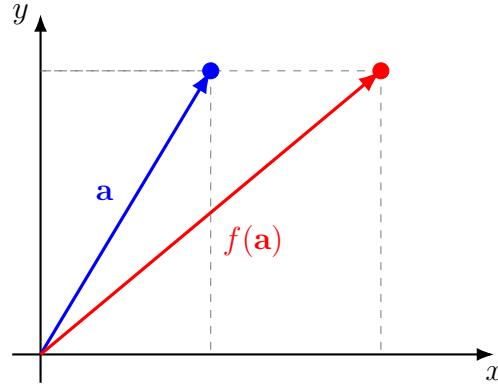
$$T(\mathbf{x}, \mathbf{y}) = (\mathbf{x} + 2\mathbf{y}, 3\mathbf{x} - \mathbf{y}) \quad (2.41)$$

adalah elemen $\mathcal{L}(\mathbb{R}^2, \mathbb{R}^2)$.

- (b) Jika $V = \mathbb{R}^3$ dan $W = \mathbb{R}^2$, maka setiap $T \in \mathcal{L}(\mathbb{R}^3, \mathbb{R}^2)$ dapat direpresentasikan oleh matriks 2×3 , misalnya

$$T(x_1, x_2, x_3) = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \quad (2.42)$$

Transformasi linear dapat divisualisasikan sebagai pemetaan yang membawa suatu vektor ke vektor lain dengan cara yang teratur (tanpa membengkokkan ruang). Sebagai contoh, Gambar 2.1 berikut memperlihatkan fungsi linear $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ dengan $f(x, y) = (2x, y)$ yang meregangkan komponen x dari setiap vektor dengan faktor 2. Vektor \mathbf{a} (biru) dipetakan ke $f(\mathbf{a})$ (merah).



Gambar 2.1 Ilustrasi transformasi linear $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ dengan $f(x, y) = (2x, y)$.
(Sumber: Dokumen penulis)

Definisi 2.1.38 (*Null Space, Range, dan Rank, (Axler, 2024)*) Untuk $T \in \mathcal{L}(V, W)$, null space dan range didefinisikan sebagai

$$\text{null } T := \{\mathbf{v} \in V : T\mathbf{v} = 0\}, \quad \text{range } T := \{T\mathbf{v} : \mathbf{v} \in V\} \subseteq W.$$

Jika V berdimensi-hingga, rank T didefinisikan oleh $\text{rank } T := \dim(\text{range } T)$. Dalam notasi matriks, untuk $\mathbf{A} \in K^{m \times n}$ yang merepresentasikan T relatif terhadap basis yang dipilih, $\text{rank}(\mathbf{A}) := \text{rank}(T) = \dim(\text{range } T)$.

Contoh 2.1.39 Misalkan

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \end{pmatrix} \in \mathbb{R}^{2 \times 3}, \quad T : \mathbb{R}^3 \rightarrow \mathbb{R}^2, \quad \text{dan} \quad T(\mathbf{x}) = \mathbf{A}\mathbf{x}.$$

(a) Sistem homogen $\mathbf{A}\mathbf{x} = 0$ memberi

$$x_2 = -4t, \quad x_1 = 5t, \quad x_3 = t, \quad (t \in \mathbb{R}),$$

sehingga

$$\text{null } T = \text{span}\{(5, -4, 1)^\top\} \quad \text{dan} \quad \dim(\text{null } T) = 1. \quad (2.43)$$

(b) Kolom pertama dan kedua bebas linear, maka

$$\text{range } T = \text{span}\{(1, 0)^\top, (2, 1)^\top\} \quad (2.44)$$

sehingga $\dim(\text{range } T) = 2$.

Contoh sebelumnya menyebutkan tentang Teorema Rank-Nulitas yang merupakan teorema fundamental dalam pemetaan linear. Lebih lanjut, secara formal, teorema tersebut dirumuskan sebagai berikut.

Teorema 2.1.40 (Teorema Fundamental Pemetaan Linear atau Rank–Nulitas, (Axler, 2024)) *Jika V berdimensi-hingga dan $T \in \mathcal{L}(V, W)$, maka*

$$\dim V = \dim(\text{null } T) + \dim(\text{range } T). \quad (2.45)$$

Contoh 2.1.41 Berdasarkan Contoh 2.1.39, diketahui bahwa $\dim(\text{null } T) = 1$ dan $\dim(\text{range } T) = 2$. Konsisten dengan Rank–Nulitas, $1 + 2 = 3 = \dim \mathbb{R}^3$.

Definisi 2.1.42 (Bentuk Multilinear Anti-Simetri, (Axler, 2024)) *Misalkan V ruang vektor atas lapangan K . Suatu fungsi $\omega : V^n \rightarrow K$ disebut bentuk n -linear jika linear pada setiap argumennya. Bentuk n -linear ω disebut anti-simetri atau alternating jika*

$$\omega(\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_j, \dots, \mathbf{v}_n) = -\omega(\mathbf{v}_1, \dots, \mathbf{v}_j, \dots, \mathbf{v}_i, \dots, \mathbf{v}_n) \quad (2.46)$$

untuk setiap pertukaran dua argumen $\mathbf{v}_i, \mathbf{v}_j \in V$. Konsekuensinya, jika ada dua argumen sama, maka $\omega(\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_i, \dots, \mathbf{v}_n) = 0$.

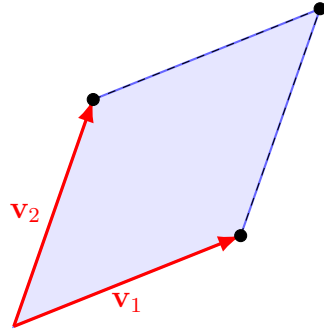
Contoh 2.1.43 Pada $V = \mathbb{R}^2$, fungsi

$$\omega((x_1, y_1), (x_2, y_2)) = x_1 y_2 - x_2 y_1 \quad (2.47)$$

adalah bentuk bilinear anti-simetri, karena

$$\omega((x_2, y_2), (x_1, y_1)) = x_2y_1 - x_1y_2 = -\omega((x_1, y_1), (x_2, y_2)). \quad (2.48)$$

Nilainya sama dengan luas terorientasi jajaran genjang yang dibentuk oleh kedua vektor.



Gambar 2.2 Ilustrasi bentuk bilinear anti-simetri $\omega(v_1, v_2) = x_1y_2 - x_2y_1$ pada \mathbb{R}^2 .
(Sumber: Dokumen penulis)

Luas dari jajar genjang pada Gambar 2.2 adalah

$$|\mathbf{v}_1||\mathbf{v}_2| \cdot \sin(\theta),$$

dengan $\cos(\theta) = \frac{\mathbf{v}_1 \mathbf{v}_2}{|\mathbf{v}_1||\mathbf{v}_2|}$. Perhatikan bahwa $\sin(\theta) = \sqrt{1 - \cos^2(\theta)}$, maka

$$\begin{aligned} |\mathbf{v}_1||\mathbf{v}_2| \sin(\theta) &= \sqrt{|\mathbf{v}_1|^2|\mathbf{v}_2|^2 - (x_1x_2 + y_1y_2)^2} \\ &= \sqrt{x_1^2x_2^2 + x_1^2y_2^2 + y_1^2x_2^2 + y_1^2y_2^2 - (x_1^2x_2^2 + 2x_1x_2y_1y_2 + y_1^2y_2^2)} \\ &= \sqrt{x_1^2y_2^2 + y_1^2x_2^2 - 2x_1x_2y_1y_2} \\ &= \sqrt{(x_1y_2 - x_2y_1)^2} \\ &= |x_1y_2 - x_2y_1|. \end{aligned} \quad (2.49)$$

Persamaan terakhir menunjukkan bahwa luas jajar genjang pada Gambar 2.2 adalah $|x_1y_2 - x_2y_1| = |\omega(v_1, v_2)|$.

Contoh 2.1.43 memberikan intuisi penting tentang determinan. Fungsi bilinear anti-simetri $\omega(v_1, v_2) = x_1y_2 - x_2y_1$ tidak lain adalah determinannya pada \mathbb{R}^2 . Secara

geometri, determinan mengukur luas terorientasi dari jajar genjang yang dibentang oleh dua vektor. Nilai absolut $|\det|$ memberikan ukuran luas (atau volume pada dimensi lebih tinggi), sedangkan tanda determinan membedakan orientasi. Jika \mathbf{v}_1 ke \mathbf{v}_2 berputar berlawanan arah jarum jam, maka $\det(\cdot) > 0$ dan sebaliknya jika searah jarum jam, maka $\det(\cdot) < 0$.

Definisi 2.1.44 (Determinan, (Axler, 2024)) Misalkan V berdimensi $n \geq 1$. Ruang semua bentuk n -linear beranti-simetri pada V memiliki dimensi 1. Untuk setiap $T \in \mathcal{L}(V)$, determinan T , ditulis $\det T$, adalah satu-satunya skalar di K dengan sifat

$$\omega(T\mathbf{v}_1, \dots, T\mathbf{v}_n) = (\det T) \omega(\mathbf{v}_1, \dots, \mathbf{v}_n), \quad (2.50)$$

untuk setiap bentuk n -linear beranti-simetri ω pada V . Jika \mathbf{A} adalah matriks $n \times n$ yang merepresentasikan T pada suatu basis V , maka determinan matriks didefinisikan oleh $\det(\mathbf{A}) := \det(T)$.

Contoh 2.1.45 Ambil $V = \mathbb{R}^2$ dan operator linear $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ yang direpresentasikan oleh matriks

$$\mathbf{A} = \begin{pmatrix} 3 & 5 \\ 7 & 2 \end{pmatrix}.$$

Untuk dua vektor $\mathbf{v}_1 = (2, 1)^\top$ dan $\mathbf{v}_2 = (1, 2)^\top$, bentuk bilinear anti-simetri standar di \mathbb{R}^2 adalah

$$\omega(\mathbf{v}_1, \mathbf{v}_2) = \det \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = 3. \quad (2.51)$$

Sekarang apabila dilihat bayangan kedua vektor tersebut,

$$T\mathbf{v}_1 = \mathbf{A}\mathbf{v}_1 = (11, 16)^\top, \quad T\mathbf{v}_2 = \mathbf{A}\mathbf{v}_2 = (13, 11)^\top.$$

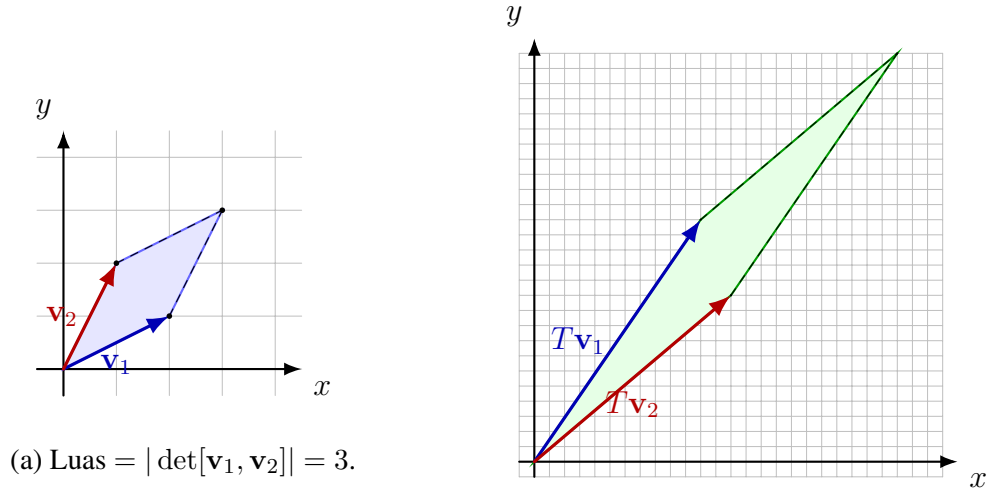
sehingga

$$\omega(T\mathbf{v}_1, T\mathbf{v}_2) = \det \begin{pmatrix} 11 & 13 \\ 16 & 11 \end{pmatrix} = 11 \cdot 11 - 13 \cdot 16 = -87. \quad (2.52)$$

Dengan demikian, sesuai definisi determinan operator,

$$\omega(T\mathbf{v}_1, T\mathbf{v}_2) = (\det T) \omega(\mathbf{v}_1, \mathbf{v}_2) \implies \det T = -87/3 = -29.$$

Sebab $\det(\mathbf{A}) := \det(T)$, diperoleh $\det(\mathbf{A}) = -29$, hasil ini konsisten dengan rumus determinan matriks 2×2 yang sudah dikenal, yaitu $\det \mathbf{A} = 6 - 35 = -29$.



(a) Luas = $|\det[\mathbf{v}_1, \mathbf{v}_2]| = 3$.

(b) Luas = $|\det[T\mathbf{v}_1, T\mathbf{v}_2]| = 87$.

Gambar 2.3 Perbandingan luas parallelogram sebelum dan sesudah transformasi A .
(Sumber: Dokumen penulis)

Berdasarkan ilustrasi di atas, terlihat bahwa $\det(\mathbf{A})$ memiliki makna geometris sebagai skala luas jajar genjang hasil transformasi oleh A .

Teorema 2.1.46 (Sifat Dasar Determinan, (Axler, 2024)) Untuk $S, T \in \mathcal{L}(V)$ dan skalar $c \in K$ berlaku:

- (a) $\det \mathbf{I}_V = 1$;
- (b) $\det(ST) = \det S \cdot \det T$; dan
- (c) T invertibel jika dan hanya jika $\det T \neq 0$.

Jika $\mathbf{A}, \mathbf{B} \in K^{n \times n}$ merepresentasikan S, T , maka $\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B})$ dan \mathbf{A} invertibel jika dan hanya jika $\det(\mathbf{A}) \neq 0$.

2.1.5. Nilai Eigen dan Vektor Eigen

Dalam aljabar linear, salah satu konsep penting adalah *eigenvalues* dan *eigenvectors* yang memberikan wawasan tentang bagaimana suatu operator linear bertindak pada vektor tertentu tanpa mengubah arahnya.

Definisi 2.1.47 (Vektor Eigen dan Nilai Eigen, (Axler, 2024, Lang, 1987)) Misalkan $T : V \rightarrow V$ adalah operator linear pada ruang vektor V atas \mathbb{C} .

(a) Suatu vektor $\mathbf{v} \in V$, $\mathbf{v} \neq \mathbf{0}$, disebut sebagai vektor eigen dari T jika terdapat skalar $\lambda \in \mathbb{C}$ sehingga

$$T(\mathbf{v}) = \lambda \mathbf{v}. \quad (2.53)$$

(b) Skalar λ yang memenuhi persamaan di atas disebut nilai eigen dari T .

Secara intuitif, vektor eigen adalah vektor yang tidak berubah arah setelah dikenakan transformasi T , melainkan hanya mengalami skala sebesar λ .

Contoh 2.1.48 Pertimbangkan $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ dengan

$$T(x, y) = (2x, 3y). \quad (2.54)$$

(a) Untuk vektor $\mathbf{v} = (1, 0)$, berlaku $T(1, 0) = (2, 0) = 2(1, 0)$, sehingga \mathbf{v} adalah vektor eigen dengan nilai $\lambda = 2$.

(b) Untuk vektor $\mathbf{w} = (0, 1)$, berlaku $T(0, 1) = (0, 3) = 3(0, 1)$, sehingga \mathbf{w} adalah vektor eigen dengan nilai $\lambda = 3$.

Teorema 2.1.49 (Keberadaan Nilai Eigen pada Ruang Vektor Kompleks, (Axler, 2024)) Jika V adalah ruang vektor berdimensi-hingga atas \mathbb{C} dan $T : V \rightarrow V$ operator linear, maka T memiliki setidaknya satu nilai eigen.

Bukti. Misalkan $\dim V = n$. Pilih basis $\mathbf{e}_1, \dots, \mathbf{e}_n$ dari V . Dalam basis ini, T direpresentasikan oleh suatu matriks $\mathbf{A} \in \text{Mat}_n(\mathbb{C})$. Selanjutnya, definisikan polinomial karakteristik

$$p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda \mathbf{I}_n). \quad (2.55)$$

Polinomial ini berderajat n dalam variabel λ , dengan koefisien dalam \mathbb{C} . Teorema dasar aljabar menyatakan bahwa setiap polinomial kompleks nonkonstan memiliki akar. Sebab $\deg p_{\mathbf{A}}(\lambda) = n \geq 1$, maka terdapat $\lambda_0 \in \mathbb{C}$ sehingga

$$p_{\mathbf{A}}(\lambda_0) = \det(\mathbf{A} - \lambda_0 \mathbf{I}_n) = 0. \quad (2.56)$$

Jika $\det(\mathbf{A} - \lambda_0 \mathbf{I}_n) = 0$, maka matriks $\mathbf{A} - \lambda_0 \mathbf{I}_n$ tidak invertibel. Hal ini berarti terdapat $\mathbf{v} \neq \mathbf{0}$ dalam \mathbb{C}^n dengan

$$(\mathbf{A} - \lambda_0 \mathbf{I}_n)\mathbf{v} = \mathbf{0}. \quad (2.57)$$

Persamaan di atas ekuivalen dengan $\mathbf{A}\mathbf{v} = \lambda_0 \mathbf{v}$. Jadi, \mathbf{v} adalah vektor eigen tak-nol dari \mathbf{A} , dan λ_0 adalah nilainya. Sebab \mathbf{A} merepresentasikan operator T , maka kesimpulan $\mathbf{A}\mathbf{v} = \lambda_0 \mathbf{v}$ identik dengan

$$T(\mathbf{v}) = \lambda_0 \mathbf{v}, \quad \mathbf{v} \neq \mathbf{0}. \quad (2.58)$$

Dengan demikian, T memiliki setidaknya satu nilai eigen di \mathbb{C} . ■

Definisi 2.1.50 (Subruang Invariant (Axler, 2024)) Subruang $U \subseteq V$ disebut *invariant terhadap T* jika $T(U) \subseteq U$. Jika \mathbf{v} adalah vektor eigen dengan nilai λ , maka $\text{span}\{\mathbf{v}\}$ adalah subruang invariant.

Contoh 2.1.51 Pertimbangkan matriks rotasi

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

Jika $\theta = 0$, semua vektor adalah vektor eigen dengan $\lambda = 1$. Jika $\theta = \pi$, semua vektor adalah vektor eigen dengan $\lambda = -1$. Jika $\theta \notin \{0, \pi\}$, maka R tidak memiliki nilai eigen real, tetapi memiliki nilai eigen kompleks $e^{i\theta}$ dan $e^{-i\theta}$.

2.1.6. Tensor

Suatu vektor \mathbf{v} memiliki elemen v_i dengan satu indeks dan suatu matriks \mathbf{M} memiliki elemen M_{ij} dengan dua indeks. Suatu tensor merupakan generalisasi dari vektor dan matriks yang memiliki $d \in \mathbb{N}$ indeks. Pembahasan klasik mengenai tensor dibahas dalam aljabar multilinear oleh Greub (1978). Pengenalan tentang tensor diawali dengan konsep pemetaan multilinear.

Definisi 2.1.52 (Pemetaan Bilinear, (Greub, 1978)) Misalkan tiga ruang vektor E, F , dan G . Definisikan suatu pemetaan

$$\varphi : E \times F \rightarrow G.$$

φ disebut *bilinear* apabila *linear terhadap masing-masing argumen ketika yang lain dianggap tetap, yaitu*

(a) *linear terhadap argumen pertama atau*

$$\varphi(\lambda x_1 + \mu x_2, y) = \lambda \varphi(x_1, y) + \mu \varphi(x_2, y), \quad x_1, x_2 \in E, y \in F, \quad (2.59)$$

dengan $\lambda, \mu \in \Gamma$ merupakan skalar; dan

(b) *linear terhadap argumen kedua atau*

$$\varphi(x, \lambda y_1 + \mu y_2) = \lambda \varphi(x, y_1) + \mu \varphi(x, y_2), \quad x \in E, y_1, y_2 \in F. \quad (2.60)$$

Apabila $G = \Gamma$, maka φ dikatakan sebagai *fungsi bilinear*.

Contoh 2.1.53 Perkalian dalam atau *dot product* merupakan salah satu contoh dari fungsi bilinear. Ambil $E = F = \mathbb{R}^2$ dan $G = \mathbb{R}$. Definisikan

$$\varphi(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \cdot \mathbf{y}, \quad (2.61)$$

yaitu *dot product*. Jika $\mathbf{x} = (x_1, x_2)$ dan $\mathbf{y} = (y_1, y_2)$, maka

$$\varphi(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + x_2 y_2. \quad (2.62)$$

Dapat ditunjukkan bahwa fungsi tersebut linear terhadap \mathbf{x} dan linear terhadap \mathbf{y} , karena

$$\begin{aligned} \varphi(\lambda \mathbf{x}^{(1)} + \mu \mathbf{x}^{(2)}, \mathbf{y}) &= \sum_{i=1}^n \left(\lambda x_i^{(1)} + \mu x_i^{(2)} \right) y_i \\ &= \sum_{i=1}^n \lambda x_i^{(1)} y_i + \sum_{i=1}^n \mu x_i^{(2)} y_i \\ &= \lambda \varphi(\mathbf{x}^{(1)}, \mathbf{y}) + \mu \varphi(\mathbf{x}^{(2)}, \mathbf{y}). \end{aligned} \quad (2.63)$$

Analog untuk kasus linear terhadap \mathbf{y} .

Contoh 2.1.54 Satu contoh fungsi bilinear lainnya adalah perkalian luar atau *outer product*. Ambil $\mathbf{x} \in \mathbb{R}^m$ dan $\mathbf{y} \in \mathbb{R}^n$. *Outer product* didefinisikan sebagai $\mathbf{x} \otimes \mathbf{y} \in \mathbb{R}^{m \times n}$ dengan elemen $(\mathbf{x} \otimes \mathbf{y})_{ij} = x_i y_j$. Pada kasus dimensi dua, $\mathbf{x} \otimes \mathbf{y} = \mathbf{xy}^\top$. Dapat diperhatikan bahwa

$$(\lambda \mathbf{x}^{(1)} + \mu \mathbf{x}^{(2)}) \otimes \mathbf{y} = \lambda(\mathbf{x}^{(1)} \otimes \mathbf{y}) + \mu(\mathbf{x}^{(2)} \otimes \mathbf{y}), \quad (2.64)$$

karena elemen dari $(\lambda \mathbf{x}^{(1)} + \mu \mathbf{x}^{(2)}) \otimes \mathbf{y}$ adalah $[(\lambda x^{(1)} + \mu x^{(2)}) \mathbf{y}]_{ij} = (\lambda x^{(1)} y + \mu x^{(2)} y)_{ij}$ yang sama dengan elemen dari $\lambda(\mathbf{x}^{(1)} \otimes \mathbf{y}) + \mu(\mathbf{x}^{(2)} \otimes \mathbf{y})$. Analog untuk kasus linear terhadap \mathbf{y} .

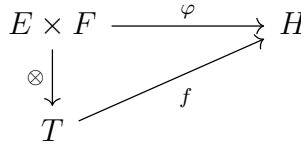
Definisi 2.1.55 (Pemetaan Multilinear, (Greub, 1978)) Misalkan terdapat $p + 1$ ruang vektor E_i ($i = 1, 2, \dots, p$) dan G . Suatu pemetaan $\varphi : E_1 \times E_2 \times \dots \times E_p \rightarrow G$ disebut sebagai *linear- p* apabila $\forall i$ ($1 \leq i \leq p$) berlaku

$$\begin{aligned} \varphi(x_1, \dots, x_{i-1}, \lambda x_i + \mu y_i, x_{i+1}, \dots, x_p) &= \lambda \varphi(x_1, \dots, x_i, \dots, x_p) \\ &\quad + \mu \varphi(x_1, \dots, y_i, \dots, y_p), \end{aligned} \quad (2.65)$$

dengan $x_i, y_i \in E$, $\lambda, \mu \in \Gamma$. Jika $G = \Gamma$, maka φ disebut sebagai fungsi linear- p .

Dalam kasus $p = 2$, subruang dari G yang dihasilkan dari vektor-vektor $\varphi(x_1, \dots, x_p)$, $x_i \in E_i$ dinotasikan sebagai $\text{Im } \varphi$.

Definisi 2.1.56 (Sifat Universal, (Greub, 1978)) Misalkan terdapat ruang vektor E dan F dan pemetaan bilinear \otimes dari $E \times F$ ke ruang vektor T . \otimes dapat dikatakan memiliki sifat universal apabila untuk setiap pemetaan bilinear $\varphi : E \times F \rightarrow H$, terdapat satu pemetaan linear yang unik $f : T \rightarrow H$ sehingga diagram di bawah ini terpenuhi.



Gambar 2.4 Diagram sifat universal.
(Sumber: Dokumen penulis)

Definisi 2.1.57 (Isomorfisme) Suatu pemetaan linear T disebut isomorfisme apabila kedua kondisi di bawah ini terpenuhi.

- (a) T bersifat satu-satu, yaitu jika $T(\mathbf{x}) = T(\mathbf{y})$, maka $\mathbf{x} = \mathbf{y}$.
- (b) T merupakan fungsi onto, yaitu jika $\mathbf{w} \in W$, maka terdapat $\mathbf{v} \in V$ sedemikian sehingga $T(\mathbf{v}) = \mathbf{w}$.

Dua subruang yang memiliki isomorfisme di atas disebut sebagai isomorfik.

Definisi 2.1.58 (Produk Tensor, (Greub, 1978)) Suatu produk tensor atau tensor product dari dua ruang vektor E dan F adalah pasangan (T, \otimes) , dengan $\otimes : E \times F \rightarrow T$ merupakan pemetaan linear dengan sifat universal. Ruang T yang secara unik ditentukan oleh E dan F hingga isomorfisme disebut sebagai produk tensor dari E dan F , dinotasikan dengan $E \otimes F$.

Definisi 2.1.59 (Ruang Tensor, (Greub, 1978; Hackbusch, 2012)) Misalkan V_1, \dots, V_d ruang vektor atas suatu lapangan K . Ruang tensor orde- d didefinisikan sebagai hasil produk tensor

$$V_1 \otimes V_2 \otimes \dots \otimes V_d,$$

yang dibangun dari kombinasi linear tensor murni

$$\mathbf{v}_1 \otimes \mathbf{v}_2 \otimes \dots \otimes \mathbf{v}_d, \quad \mathbf{v}_i \in V_i.$$

Teorema 2.1.60 (Sifat Universal Produk Tensor, (Greub, 1978)) Diberikan pemetaan multilinear

$$\varphi : V_1 \times \dots \times V_d \rightarrow W,$$

maka terdapat pemetaan linear unik $\tilde{\varphi} : V_1 \otimes \dots \otimes V_d \rightarrow W$ sedemikian sehingga

$$\varphi(\mathbf{v}_1, \dots, \mathbf{v}_d) = \tilde{\varphi}(\mathbf{v}_1 \otimes \dots \otimes \mathbf{v}_d). \quad (2.66)$$

Contoh 2.1.61 Berikut ini adalah contoh untuk beberapa dimensi d .

1. Untuk $d = 1$, diperoleh V_1 , sehingga tensor orde-1 identik dengan vektor.
2. Untuk $d = 2$, diperoleh $V_1 \otimes V_2$, yang isomorfik dengan ruang matriks $\dim(V_1) \times \dim(V_2)$. Matriks adalah tensor orde-2.
3. Untuk $d = 3$ dan $V = \mathbb{R}^n$, setiap $\mathcal{T} \in V^{\otimes 3}$ ditulis

$$\mathcal{T} = \sum_{i,j,k=1}^n t_{ijk} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k,$$

dengan \mathbf{e}_i basis standar dan $t_{ijk} \in \mathbb{R}$.

Definisi 2.1.62 (Basis dan Dimensi Tensor, (Hackbusch, 2012)) Jika $\{\mathbf{e}_i\}_{i=1}^n$ basis V , maka

$$\{ \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \otimes \dots \otimes \mathbf{e}_{i_d} : 1 \leq i_1, \dots, i_d \leq n \}$$

adalah basis dari $V^{\otimes d}$, dan

$$\dim(V^{\otimes d}) = (\dim V)^d. \quad (2.67)$$

Definisi 2.1.63 (Tensor Murni dan Rank, (Hackbusch, 2012)) Sebuah tensor $\mathcal{X} \in V_1 \otimes \cdots \otimes V_d$ disebut murni jika

$$\mathcal{X} = \mathbf{v}_1 \otimes \cdots \otimes \mathbf{v}_d.$$

Tensor umum dapat ditulis sebagai kombinasi linear hingga

$$\mathcal{X} = \sum_{r=1}^R \mathbf{v}_1^{(r)} \otimes \cdots \otimes \mathbf{v}_d^{(r)},$$

dan R minimum yang mungkin disebut rank dari \mathcal{X} .

Contoh 2.1.64 Ambil $V = \mathbb{R}^2$ dengan basis $\mathbf{e}_1, \mathbf{e}_2$. Tensor

$$\mathcal{X} = \mathbf{e}_1 \otimes \mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1 \otimes \mathbf{e}_1$$

memerlukan dua suku murni, sehingga $\text{rank}(\mathcal{X}) = 2$.

Definisi 2.1.65 (Matrisasi dan j -rank, (Hackbusch, 2012)) Misalkan $\mathcal{X} \in \bigotimes_{j=1}^d K^{I_j}$ adalah tensor orde- d dengan ukuran $I_1 \times I_2 \times \cdots \times I_d$. Matrisasi mode- j dari \mathcal{X} , ditulis $M_j(\mathcal{X})$, adalah matriks dalam $K^{I_j \times I_{[j]}}$ dengan $I_{[j]} = \prod_{k \neq j} I_k$, yang diperoleh dengan meratakan tensor sepanjang mode- j . Dengan kata lain, indeks i_j dibiarkan tetap sebagai indeks baris, sedangkan gabungan semua indeks lainnya $(i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_d)$ dipetakan menjadi indeks kolom. Rank dari matriks $M_j(\mathcal{X})$ disebut j -rank dari \mathcal{X} .

Contoh 2.1.66 Ambil tensor orde-3 $\mathcal{T} \in \mathbb{R}^{2 \times 3 \times 4}$. Matrisasi mode-1 menghasilkan matriks $M_1(\mathcal{T}) \in \mathbb{R}^{2 \times 12}$, dengan baris menyatakan dimensi pertama ($i_1 = 1, 2$) dan kolom merupakan penggabungan dari indeks dimensi kedua ($i_2 = 1, 2, 3$) dan ketiga ($i_3 = 1, \dots, 4$). Secara skematis, tensor berbentuk balok $2 \times 3 \times 4$ diubah

menjadi matriks berukuran 2×12 .

Definisi 2.1.67 (Perkalian Kronecker, (Hackbusch, 2012)) Untuk $\mathbf{A} \in \text{Mat}_{m \times n}$ dan $\mathbf{B} \in \text{Mat}_{p \times q}$, perkalian Kronecker dari matriks tersebut dituliskan sebagai $\mathbf{A} \otimes \mathbf{B}$, didefinisikan sebagai

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \in \text{Mat}_{mp \times nq}. \quad (2.68)$$

Contoh 2.1.68 Apabila diambil

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \in \mathbb{R}^{2 \times 2}, \quad \mathbf{B} = \begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} \in \mathbb{R}^{2 \times 2}.$$

Perkalian Kronecker $\mathbf{A} \otimes \mathbf{B}$ adalah

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} 1 \cdot \mathbf{B} & 2 \cdot \mathbf{B} \\ 3 \cdot \mathbf{B} & 4 \cdot \mathbf{B} \end{pmatrix} = \begin{pmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{pmatrix}. \quad (2.69)$$

Hasilnya adalah matriks berukuran 4×4 karena \mathbf{A} berukuran 2×2 dan \mathbf{B} juga 2×2 , sehingga ukurannya $2 \cdot 2 \times 2 \cdot 2 = 4 \times 4$.

2.1.7. Dekomposisi Matriks dan Tensor

Dekomposisi matriks merupakan representasi suatu matriks sebagai hasil kali atau gabungan matriks-matriks lain yang bentuknya lebih sederhana dan terstruktur. Seiring berkembangnya analisis data multidimensi, konsep dekomposisi meluas dari matriks ke tensor. Oleh karena itu, pemahaman yang sistematis terhadap dekomposisi matriks dan tensor menjadi esensial dalam penelitian ini. Namun, sebelum membahas masing-masing dekomposisi, bagian berikut memberikan definisi-definisi yang digunakan dalam proses dekomposisi.

Definisi 2.1.69 (Hasil Kali Dalam, (Axler, 2024)) Misalkan V ruang vektor atas F ($F = \mathbb{R}$ atau $F = \mathbb{C}$). Suatu fungsi $\langle \cdot, \cdot \rangle : V \times V \rightarrow F$ disebut hasil kali dalam (inner product) jika untuk semua $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ dan $\lambda \in F$ berlaku

$$(a) \langle \mathbf{v}, \mathbf{v} \rangle \geq 0;$$

$$(b) \langle \mathbf{v}, \mathbf{v} \rangle = 0 \iff \mathbf{v} = \mathbf{0};$$

$$(c) \langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle, \text{ dan } \langle \lambda \mathbf{u}, \mathbf{v} \rangle = \lambda \langle \mathbf{u}, \mathbf{v} \rangle; \text{ dan}$$

$$(d) \langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}.$$

Definisi 2.1.70 (Adjoint, (Axler, 2024)) Untuk operator linear $T : V \rightarrow W$ antar ruang berhasil kali dalam, adjoint $T^* : W \rightarrow V$ didefinisikan sebagai pemetaan unik yang memenuhi

$$\langle T\mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{v}, T^*\mathbf{w} \rangle \quad (2.70)$$

untuk semua $\mathbf{v} \in V, \mathbf{w} \in W$.

Definisi 2.1.71 (Matriks Unitary dan Ortogonal, (Axler, 2024)) Matriks persegi $\mathbf{Q} \in \mathbb{C}^{n \times n}$ disebut unitary jika $\mathbf{Q}^* \mathbf{Q} = \mathbf{I}$. Jika \mathbf{Q} real, syarat ini setara dengan $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$, dan \mathbf{Q} disebut ortogonal.

Definisi 2.1.72 (Matriks Hermitian, (Axler, 2024)) Matriks $\mathbf{A} \in \mathbb{C}^{n \times n}$ disebut Hermitian jika $\mathbf{A}^* = \mathbf{A}$. Jika \mathbf{A} real, syarat ini setara dengan $\mathbf{A}^\top = \mathbf{A}$ atau simetris.

Definisi 2.1.73 (Dekomposisi Eigen, (Axler, 2024)) Misalkan $\mathbf{A} \in \mathbb{C}^{n \times n}$. Jika \mathbf{A} memiliki n buah vektor eigen bebas linear, maka matriks \mathbf{A} disebut diagonalisabel. Dalam hal ini, terdapat matriks invertibel \mathbf{P} yang kolom-kolomnya merupakan vektor-vektor eigen \mathbf{A} , serta matriks diagonal $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$ yang berisi nilai-nilai eigen λ_i pada diagonal utamanya, sehingga

$$\mathbf{A} = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^{-1}. \quad (2.71)$$

Faktorisasi ini disebut dekomposisi eigen atau eigendecomposition.

Contoh 2.1.74 Pertimbangkan matriks

$$\mathbf{A} = \begin{pmatrix} 5 & 4 \\ 1 & 2 \end{pmatrix}.$$

Karakteristik polinomial dari \mathbf{A} adalah $\det(\mathbf{A} - \lambda I) = \lambda^2 - 7\lambda + 6$, sehingga diperoleh nilai eigen $\lambda_1 = 6$ dan $\lambda_2 = 1$. Untuk $\lambda_1 = 6$, persamaan $(\mathbf{A} - 6I)\mathbf{v} = 0$ menghasilkan vektor eigen $\mathbf{v}_1 = (4, 1)^\top$. Sementara itu, untuk $\lambda_2 = 1$, persamaan $(\mathbf{A} - I)\mathbf{v} = 0$ memberikan vektor eigen $\mathbf{v}_2 = (-1, 1)^\top$. Dengan demikian, dapat dibentuk matriks

$$\mathbf{P} = \begin{pmatrix} 4 & -1 \\ 1 & 1 \end{pmatrix},$$

yang kolom-kolomnya adalah vektor-vektor eigen \mathbf{A} , serta matriks diagonal

$$\mathbf{\Lambda} = \begin{pmatrix} 6 & 0 \\ 0 & 1 \end{pmatrix},$$

yang berisi nilai-nilai eigen pada diagonal utamanya. Oleh karena itu, \mathbf{A} dapat dituliskan dalam bentuk

$$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1},$$

yang merupakan dekomposisi eigen dari matriks \mathbf{A} . Dalam hal ini, dekomposisi menunjukkan bahwa transformasi linear yang direpresentasikan oleh \mathbf{A} ekuivalen dengan transformasi basis ke arah vektor-vektor eigen, dilanjutkan dengan transformasi diagonal, kemudian dikembalikan lagi ke basis semula melalui \mathbf{P}^{-1} .

Definisi 2.1.75 (Faktorisasi QR, (Axler, 2024; Hackbusch, 2012)) Untuk setiap $\mathbf{A} \in \mathbb{F}^{m \times n}$, terdapat matriks unitary atau ortogonal \mathbf{Q} dan segitiga atas \mathbf{R} sehingga

$$\mathbf{A} = \mathbf{Q}\mathbf{R}. \quad (2.72)$$

QR dapat dikonstruksi melalui Algoritma Gram–Schmidt atau Householder. Pemakaian Gram–Schmidt dan penyelesaian $\mathbf{A}\mathbf{x} = \mathbf{b}$ dengan $\mathbf{R}\mathbf{x} = \mathbf{Q}^\mathbf{b}$ dibahas di*

(Axler, 2024). Kompleksitas dan varian numeriknya, yaitu pivoted *QR* dan *Householder* dirangkum di (Hackbusch, 2012).

Contoh 2.1.76 Pertimbangkan matriks

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

dengan kolom-kolom $\mathbf{a}_1 = (1, 1, 0)^\top$ dan $\mathbf{a}_2 = (1, 0, 1)^\top$. Prosedur Gram–Schmidt dimulai dengan menormalkan kolom pertama, sehingga

$$\|\mathbf{a}_1\| = \sqrt{1^2 + 1^2 + 0^2} = \sqrt{2}, \quad \mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|} = \frac{1}{\sqrt{2}}(1, 1, 0)^\top.$$

Selanjutnya, kolom kedua diortogonalisasi terhadap \mathbf{q}_1 melalui proyeksi; koefisien proyeksinya adalah

$$r_{12} = \langle \mathbf{q}_1, \mathbf{a}_2 \rangle = \frac{1}{\sqrt{2}}(1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1) = \frac{1}{\sqrt{2}},$$

sehingga vektor residu menjadi

$$\hat{\mathbf{a}}_2 = \mathbf{a}_2 - r_{12}\mathbf{q}_1 = (1, 0, 1)^\top - \frac{1}{2}(1, 1, 0)^\top = \left(\frac{1}{2}, -\frac{1}{2}, 1\right)^\top.$$

Normalisasi residu tersebut menghasilkan

$$\|\hat{\mathbf{a}}_2\| = \sqrt{\left(\frac{1}{2}\right)^2 + \left(-\frac{1}{2}\right)^2 + 1^2} = \sqrt{\frac{3}{2}}, \quad \mathbf{q}_2 = \frac{\hat{\mathbf{a}}_2}{\|\hat{\mathbf{a}}_2\|} = \left(\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}\right)^\top.$$

Dengan demikian, kolom-kolom ortonormal terkumpul dalam

$$\mathbf{Q} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{pmatrix}.$$

Matriks segitiga atas \mathbf{R} diperoleh dari koefisien Gram–Schmidt, yakni $r_{11} = \|\mathbf{a}_1\| = \sqrt{2}$, $r_{12} = \langle \mathbf{q}_1, \mathbf{a}_2 \rangle = \frac{1}{\sqrt{2}}$, dan $r_{22} = \|\hat{\mathbf{a}}_2\| = \sqrt{\frac{3}{2}}$, dengan $r_{21} = 0$, sehingga

$$\mathbf{R} = \begin{pmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{\frac{3}{2}} \end{pmatrix}.$$

Terlihat bahwa $\mathbf{A} = \mathbf{QR}$ dengan \mathbf{Q} berkolom ortonormal dan \mathbf{R} segitiga atas, sesuai definisi faktorisasi QR.

Definisi 2.1.77 (Dekomposisi Nilai Tunggal atau *Singular Value Decomposition* (SVD), (Axler, 2024; Hackbusch, 2012)) Untuk $\mathbf{A} \in \mathbb{F}^{m \times n}$ terdapat unitary atau ortogonal $\mathbf{U} \in \mathbb{F}^{m \times m}$, $\mathbf{V} \in \mathbb{F}^{n \times n}$, dan diagonal taknegatif $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ sehingga

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \quad (2.73)$$

dengan nilai singular $\sigma_1 \geq \dots \geq \sigma_r > 0$.

Contoh 2.1.78 (SVD pada matriks 2×2) Pertimbangkan matriks

$$\mathbf{A} = \begin{pmatrix} 3 & 0 \\ 4 & 5 \end{pmatrix}.$$

Perhitungan SVD diawali dengan membentuk $\mathbf{A}^\top \mathbf{A}$, yaitu

$$\mathbf{A}^\top \mathbf{A} = \begin{pmatrix} 3 & 4 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 4 & 5 \end{pmatrix} = \begin{pmatrix} 25 & 20 \\ 20 & 25 \end{pmatrix}.$$

Matriks simetris ini memiliki polinomial karakteristik

$$\det \left(\begin{pmatrix} 25 - \lambda & 20 \\ 20 & 25 - \lambda \end{pmatrix} \right) = (25 - \lambda)^2 - 400 = 0,$$

sehingga nilai eigennya adalah $\lambda_1 = 45$ dan $\lambda_2 = 5$. Nilai singular σ_i diperoleh

sebagai akar kuadrat positif dari nilai eigen tersebut, yakni

$$\sigma_1 = \sqrt{45} = 3\sqrt{5}, \quad \sigma_2 = \sqrt{5}.$$

Vektor-vektor singular kanan adalah eigenvektor dari $\mathbf{A}^\top \mathbf{A}$. Untuk $\lambda_1 = 45$ persamaan $(\mathbf{A}^\top \mathbf{A} - 45\mathbf{I})\mathbf{v} = \mathbf{0}$ memberikan ruang eigen yang dibentang oleh $(1, 1)^\top$, sedangkan untuk $\lambda_2 = 5$ memberikan ruang eigen yang dibentang oleh $(1, -1)^\top$. Dengan mengortonormalkan keduanya, dapat diambil

$$\mathbf{V} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Selanjutnya, vektor-vektor singular kiri ditentukan dari rumus $\mathbf{u}_i = \frac{1}{\sigma_i} \mathbf{A} \mathbf{v}_i$. Substitusi menghasilkan

$$\mathbf{u}_1 = \frac{1}{3\sqrt{5}} \mathbf{A} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right) = \begin{pmatrix} \frac{1}{\sqrt{10}} \\ \frac{3}{\sqrt{10}} \end{pmatrix}, \quad \mathbf{u}_2 = \frac{1}{\sqrt{5}} \mathbf{A} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right) = \begin{pmatrix} \frac{3}{\sqrt{10}} \\ -\frac{1}{\sqrt{10}} \end{pmatrix},$$

yang keduanya saling ortonormal. Dengan demikian,

$$\mathbf{U} = \begin{pmatrix} \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \\ \frac{3}{\sqrt{10}} & -\frac{1}{\sqrt{10}} \end{pmatrix}, \quad \mathbf{\Sigma} = \begin{pmatrix} 3\sqrt{5} & 0 \\ 0 & \sqrt{5} \end{pmatrix}, \quad \mathbf{V} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

sehingga faktorisasi nilai tunggal $\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$ terbentuk. Interpretasi faktorisasi ini adalah bahwa transformasi linear yang direpresentasikan oleh \mathbf{A} ekuivalen dengan perubahan basis ke arah vektor-vektor singular kanan, diikuti oleh penskalaan diagonal oleh nilai-nilai singular, lalu dilanjutkan dengan pemetaan kembali melalui basis vektor singular kiri.

Definisi 2.1.79 (Dekomposisi Cholesky, (Axler, 2024; Hackbusch, 2012)) Jika $\mathbf{A} \in \mathbb{R}^{n \times n}$ simetris dan positif definit, maka terdapat unik segitiga bawah \mathbf{L} dengan diagonal positif sehingga $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ (atau $\mathbf{A} = \mathbf{R}^* \mathbf{R}$ dengan \mathbf{R} segitiga atas).

Contoh 2.1.80 (Dekomposisi Cholesky pada matriks 2×2) Pertimbangkan matriks simetris positif definit

$$\mathbf{A} = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}.$$

Sebab determinan minor utamanya bernilai positif, yakni $\det(4) = 4 > 0$ dan $\det(\mathbf{A}) = 4 \cdot 3 - 2 \cdot 2 = 8 > 0$, maka \mathbf{A} memang positif definit. Dekomposisi Cholesky mencari segitiga bawah $\mathbf{L} = \begin{pmatrix} \ell_{11} & 0 \\ \ell_{21} & \ell_{22} \end{pmatrix}$ dengan diagonal positif sehingga $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$. Dengan menyamakan entri pada

$$\mathbf{L}\mathbf{L}^\top = \begin{pmatrix} \ell_{11}^2 & \ell_{11}\ell_{21} \\ \ell_{11}\ell_{21} & \ell_{21}^2 + \ell_{22}^2 \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix},$$

diperoleh berturut-turut $\ell_{11}^2 = 4$ sehingga $\ell_{11} = 2$, $\ell_{11}\ell_{21} = 2$ sehingga $\ell_{21} = 1$, dan $\ell_{21}^2 + \ell_{22}^2 = 3$ sehingga $\ell_{22} = \sqrt{2}$. Dengan demikian,

$$\mathbf{L} = \begin{pmatrix} 2 & 0 \\ 1 & \sqrt{2} \end{pmatrix} \quad \text{dan} \quad \mathbf{A} = \mathbf{L}\mathbf{L}^\top.$$

Untuk tensor, dinotasikan tensor orde-3 sebagai $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$. Mode- n product ditulis $\mathcal{X} \times_n \mathbf{A}$, yaitu .

Definisi 2.1.81 (Dekomposisi CP dan Tucker, (Hackbusch, 2012)) Dua pendekatan yang paling umum digunakan untuk mendekomposisi tensor berorde lebih tinggi adalah dekomposisi CP (Canonical Polyadic atau CANDECOMP) dan dekomposisi Tucker atau High-Order Singular Value Decomposition (HOSVD).

Dekomposisi CP bertujuan menuliskan suatu tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ sebagai jumlah terbatas dari tensor dengan rank satu, yaitu

$$\mathcal{X} \approx \sum_{r=1}^R \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r, \quad (2.74)$$

dengan $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, dan $\mathbf{c}_r \in \mathbb{R}^K$. Jumlah komponen R disebut rank CP

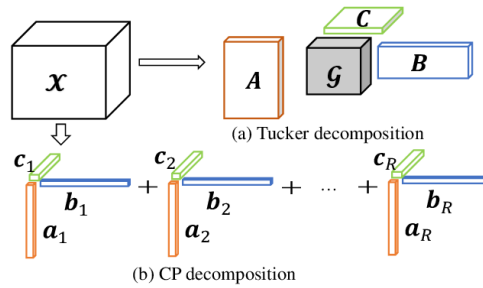
dari tensor, dan dekomposisi ini sering dipandang sebagai generalisasi langsung dari dekomposisi eigen untuk matriks, karena setiap komponen merupakan analogi vektor eigen yang membentuk rank-one tensor.

Sementara itu, dekomposisi Tucker atau HOSVD menuliskan tensor \mathcal{X} dalam bentuk

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}, \quad (2.75)$$

dengan \mathcal{G} merupakan tensor inti yang berukuran lebih kecil dan $\mathbf{A}, \mathbf{B}, \mathbf{C}$ adalah matriks faktor ortonormal. Tidak seperti CP yang memaksa representasi dalam bentuk jumlah rank satu, dekomposisi Tucker memperbolehkan adanya interaksi yang lebih kompleks di dalam tensor inti \mathcal{G} . HOSVD menjamin bahwa matriks faktor diperoleh dari dekomposisi nilai tunggal (SVD) pada matriks hasil pembentangan (unfolding) tensor sepanjang setiap mode. Dekomposisi ini juga dikenal bersifat quasi-optimal, dalam arti bahwa jika dilakukan pemotongan (truncation) pada singular values, maka kesalahan aproksimasi yang diperoleh tetap terkendali secara teoretis.

Perhatikan ilustrasi di bawah ini untuk penjelasan terkait dekomposisi CP dan Tucker secara visual.



Gambar 2.5 Ilustrasi (a) Dekomposisi Tucker dan (b) Dekomposisi CP pada Tensor. (Sumber: Jiang dkk., 2020)

Contoh 2.1.82 Sebagai ilustrasi, ambil tensor $\mathcal{X} \in \mathbb{R}^{2 \times 2 \times 2}$ dengan entri

$$\mathcal{X}_{ijk} = \begin{cases} 1, & \text{jika } i = j = k, \\ 0, & \text{selainnya.} \end{cases}$$

Tensor ini hanya bernilai 1 pada posisi diagonal $(1, 1, 1)$ dan $(2, 2, 2)$, sedangkan seluruh elemen lainnya bernilai 0.

Dalam kerangka CP, tensor ini dapat direpresentasikan sebagai jumlah dari dua rank-one tensors:

$$\mathcal{X} \approx \mathbf{e}_1 \otimes \mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes \mathbf{e}_2 \otimes \mathbf{e}_2,$$

dengan $\mathbf{e}_1 = (1, 0)^\top$ dan $\mathbf{e}_2 = (0, 1)^\top$. Setiap komponen di sini langsung menangkap satu “garis diagonal” dari tensor, sehingga rank CP-nya adalah 2. Hal ini mirip dengan menyatakan sebuah matriks diagonal sebagai jumlah dari matriks-matriks *rank* satu.

Dalam kerangka Tucker/HOSVD, prosedur dilakukan dengan membentangkan tensor \mathcal{X} ke dalam tiga matriks *unfolding* sesuai mode-1, mode-2, dan mode-3. Masing-masing matriks kemudian dikenai dekomposisi nilai tunggal (SVD) sehingga diperoleh faktor ortonormal \mathbf{A} , \mathbf{B} , dan \mathbf{C} . Tensor inti \mathcal{G} diperoleh dengan mengalikan \mathcal{X} berturut-turut dengan \mathbf{A}^\top , \mathbf{B}^\top , dan \mathbf{C}^\top pada mode yang sesuai. Dalam contoh ini, faktor-faktor ortonormal yang muncul berkorespondensi langsung dengan basis standar \mathbf{e}_1 , \mathbf{e}_2 , sedangkan inti \mathcal{G} akan berukuran $2 \times 2 \times 2$ dengan entri yang lebih terkonsentrasi pada diagonal.

2.1.8. Diferensial Vektor dan Matriks

Kalkulus matriks adalah perpanjangan dari kalkulus diferensial pada ruang Euclidean ke fungsi yang melibatkan vektor dan matriks sebagai argumen maupun hasil.

Definisi 2.1.83 (Konvensi Turunan pada Vektor dan Matriks, (Dhrymes, 2013))

Misalkan

$$\mathbf{y} = \psi(\mathbf{x}), \quad (2.76)$$

dengan \mathbf{y} dan \mathbf{x} secara berurutan merupakan vektor kolom berukuran m dan n .

Notasi

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \left[\frac{\partial y_i}{\partial x_j} \right], \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \quad (2.77)$$

merupakan notasi dari matriks derivatif parsial orde pertama (atau matriks Jacobi-an) dari transformasi \mathbf{x} ke \mathbf{y} sedemikian sehingga baris ke- i berisi turunan parsial dari y_i terhadap elemen dari \mathbf{x} , yaitu

$$\frac{\partial y_i}{\partial x_1}, \frac{\partial y_i}{\partial x_2}, \dots, \frac{\partial y_i}{\partial x_n}. \quad (2.78)$$

Proposisi 2.1.84 *Jika*

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (2.79)$$

dengan \mathbf{A} merupakan matriks berukuran $m \times n$ yang tidak bergantung pada \mathbf{x} , maka berlaku

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{A}. \quad (2.80)$$

Proposisi 2.1.85 *Jika*

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad (2.81)$$

dengan \mathbf{y} berukuran $m \times 1$, \mathbf{A} berukuran $m \times n$, \mathbf{x} berukuran $n \times 1$, serta \mathbf{A} dan \mathbf{x} bergantung pada vektor $\boldsymbol{\alpha}$ berukuran r , maka

$$\frac{\partial \mathbf{y}}{\partial \boldsymbol{\alpha}} = (\mathbf{x}^\top \otimes \mathbf{I}_m) \frac{\partial \mathbf{A}}{\partial \boldsymbol{\alpha}} + \mathbf{A} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\alpha}}. \quad (2.82)$$

Proposisi 2.1.86 *Jika*

$$\mathbf{y} = \mathbf{z}^\top \mathbf{A}\mathbf{x}, \quad (2.83)$$

dengan \mathbf{z} berukuran $m \times 1$, \mathbf{A} berukuran $m \times n$, \mathbf{x} berukuran $n \times 1$, serta \mathbf{A} independen dari \mathbf{z} dan \mathbf{x} , maka

$$\frac{\partial \mathbf{y}}{\partial \mathbf{z}} = \mathbf{x}^\top \mathbf{A}^\top, \quad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{z}^\top \mathbf{A}. \quad (2.84)$$

Proposisi 2.1.87 *Jika*

$$\mathbf{y} = \mathbf{x}^\top \mathbf{A}\mathbf{x}, \quad (2.85)$$

dengan \mathbf{x} berukuran $n \times 1$ serta \mathbf{A} berukuran $n \times n$ dan independen dengan \mathbf{x} ,

maka

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{x}^\top (\mathbf{A} + \mathbf{A}^\top). \quad (2.86)$$

Jika \mathbf{A} adalah matriks simetri, maka

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = 2\mathbf{x}^\top \mathbf{A}. \quad (2.87)$$

Proposisi 2.1.88 Misalkan matriks \mathbf{A} nonsingular yang berukuran $m \times m$ dan bergantung dengan parameter skalar α , maka

$$\frac{\partial \mathbf{A}^{-1}}{\partial \alpha} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \alpha} \mathbf{A}^{-1}. \quad (2.88)$$

Teorema 2.1.89 (Aturan Jejak, , (Dhrymes, 2013)) Untuk matriks dengan ukuran yang sesuai berlaku:

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{A}\mathbf{X}) = \mathbf{A}^\top, \quad \frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{X}^\top \mathbf{A}\mathbf{X}) = (\mathbf{A} + \mathbf{A}^\top)\mathbf{X}.$$

2.2. Regresi Spasial dan Spasial-Temporal

Analisis regresi merupakan suatu teknik statistik untuk menginvestigasi dan memodelkan hubungan antarvariabel. Aplikasi dari analisis regresi sangat luas dan digunakan dalam hampir semua bidang, seperti teknik atau *engineering*, ilmu fisika dan kimia, ekonomi, manajemen, ilmu biologi, dan sosiologi. Dalam analisis regresi, variabel respons atau variabel dependen (dinotasikan dengan variabel random Y) dimodelkan dengan fungsi dalam variabel penjelas atau variabel independen (dinotasikan dengan variabel random X). (Montgomery, Peck, & Vining, 2012)

2.2.1. Regresi Linear Biasa atau *Ordinary Least Squares* (OLS)

Analisis regresi linear memodelkan hubungan antarvariabel dengan fungsi linear. Apabila terdapat satu variabel prediktor maka model yang terbentuk disebut sebagai regresi linear sederhana, sedangkan apabila terdapat lebih dari satu variabel prediktor maka model yang terbentuk disebut sebagai regresi linear berganda.

Definisi 2.2.1 (Regresi Linear Sederhana, (Montgomery dkk., 2012)) Model regresi linear sederhana merupakan model dengan satu variabel penjelas X yang mempunyai hubungan garis lurus terhadap satu respons Y . Model ini dirumuskan dengan

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x} + \boldsymbol{\varepsilon}, \quad (2.89)$$

dengan intersep β_0 dan koefisien β_1 merupakan konstanta yang tidak diketahui, serta $\boldsymbol{\varepsilon}$ adalah vektor galat acak. \mathbf{y} dan \mathbf{x} berturut-turut adalah vektor acak Y dan X . Parameter β_0 dan β_1 biasa disebut sebagai koefisien-koefisien regresi.

Definisi 2.2.2 (Regresi Linear Berganda, (Montgomery dkk., 2012)) Regresi linear berganda merupakan model regresi yang memiliki lebih dari satu variabel independen. Model ini merupakan generalisasi dari regresi linear sederhana. Vektor respons \mathbf{y} mungkin memiliki hubungan dengan p -buah variabel penjelas, yaitu matriks acak \mathbf{X} yang dibentuk atas p -buah vektor acak $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p$, sehingga

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \dots + \beta_p \mathbf{x}_p + \boldsymbol{\varepsilon}. \quad (2.90)$$

Model di atas dapat dituliskan dalam notasi matriks dan vektor, yaitu

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1p} \\ 1 & X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}, \quad (2.91)$$

atau

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \quad (2.92)$$

Model OLS mengasumsikan bahwa galat memiliki rata-rata nol dan variansi konstan σ^2 yang tidak diketahui nilainya. Lebih lanjut, model ini juga mengasumsikan bahwa komponen galat tidak memiliki autokorelasi yang berarti bahwa nilai dari suatu galat tidak bergantung pada galat lainnya. Dengan kata lain $\boldsymbol{\varepsilon} \stackrel{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2)$, untuk suatu σ^2 yang bersifat konstan.

Model regresi linear memandang vektor acak \mathbf{y} sebagai fungsi dari matriks acak \mathbf{X} . Oleh karena itu, rata-rata dari distribusi \mathbf{y} dapat dituliskan sebagai

$$\mathbb{E}[\mathbf{y} \mid \mathbf{X}] = \mathbb{E}[\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}] = \mathbf{X}\boldsymbol{\beta} \quad (2.93)$$

dan variansinya

$$\text{Var}(\mathbf{y} \mid \mathbf{X}) = \text{Var}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}) = \sigma^2 \mathbf{I}_n, \quad (2.94)$$

karena $\mathbb{E}[\boldsymbol{\varepsilon}] = \mathbf{0}$ dan $\text{Var}(\boldsymbol{\varepsilon}) = \sigma^2 \mathbf{I}_n$. Oleh karena itu, rata-rata dari \mathbf{y} adalah fungsi linear dari \mathbf{X} meskipun variansinya tidak bergantung pada nilai \mathbf{X} .

Definisi 2.2.3 (Asumsi Regresi Linear Klasik, (Greene, 2002; Gujarati & Porter, 2009; Montgomery dkk., 2012)) *Model regresi linear klasik didasarkan pada sejumlah asumsi yang dikenal sebagai asumsi Gauss–Markov. Asumsi-asumsi ini diperlukan agar penduga kuadrat terkecil memiliki sifat optimal sebagai best linear unbiased estimator (BLUE). Asumsi-asumsi tersebut adalah sebagai berikut.*

- (a) *Model bersifat linear dalam parameter, yaitu $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$.*
- (b) *Nilai tengah galat adalah nol bersyarat pada \mathbf{X} , yaitu $\mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$.*
- (c) *Matriks desain \mathbf{X} memiliki rank penuh sehingga $\mathbf{X}^\top \mathbf{X}$ nonsingular.*
- (d) *Galat memiliki ragam konstan, $\text{Var}(\varepsilon_i \mid \mathbf{X}) = \sigma^2$ untuk semua i .*
- (e) *Komponen galat tidak berkorelasi, $\text{Cov}(\varepsilon_i, \varepsilon_j \mid \mathbf{X}) = 0$ untuk $i \neq j$.*
- (f) *Untuk keperluan inferensi, diasumsikan $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$.*

Estimasi parameter dalam regresi linear dilakukan dengan metode kuadrat terkecil atau *ordinary least squares* (OLS). Prinsip dasarnya adalah mencari nilai $\boldsymbol{\beta}$ yang meminimumkan jumlah kuadrat galat

$$S(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (2.95)$$

Perhatikan bahwa fungsi kuadrat terkecil dari persamaan di atas dapat dituliskan

sebagai

$$S(\beta) = (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) = \mathbf{y}^\top \mathbf{y} - 2\beta^\top \mathbf{X}^\top \mathbf{y} + \beta^\top \mathbf{X}^\top \mathbf{X} \beta. \quad (2.96)$$

Dengan diferensial matriks, gradien terhadap β adalah

$$\nabla_\beta S(\beta) = -2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X} \beta. \quad (2.97)$$

Syarat orde pertama meminimumkan S adalah $\nabla_\beta S(\beta) = \mathbf{0}$, sehingga diperoleh

$$\mathbf{X}^\top \mathbf{X} \hat{\beta} = \mathbf{X}^\top \mathbf{y}, \quad (2.98)$$

yang disebut persamaan normal. Selanjutnya, matriks Hessian atau orde kedua adalah

$$\nabla_\beta^2 S(\beta) = 2\mathbf{X}^\top \mathbf{X} \succeq \mathbf{0}, \quad (2.99)$$

sehingga S konveks dalam β . Jika $\mathbf{X}^\top \mathbf{X}$ non-singular atau setara $\text{rank}(\mathbf{X}) = p$, maka Hessian definit positif dan solusi orde pertama unik serta memberikan minimum global.

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \quad (2.100)$$

Jika $\mathbf{X}^\top \mathbf{X}$ singular, himpunan minimum tidak tunggal. Salah satu solusi adalah solusi *norma-minimum* $\hat{\beta} = \mathbf{X}^+ \mathbf{y}$ dengan \mathbf{X}^+ merupakan *Moore–Penrose pseudoinverse*.

Teorema 2.2.4 (Teorema Gauss–Markov, (Greene, 2002; Montgomery dkk., 2012)) *Di bawah asumsi regresi linear klasik, penduga OLS $\hat{\beta}$ bersifat tak bias, yaitu $\mathbb{E}[\hat{\beta}] = \beta$, dan memiliki kovariansi*

$$\text{Var}(\hat{\beta}) = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}. \quad (2.101)$$

Selain itu, $\hat{\beta}$ merupakan best linear unbiased estimator (BLUE), artinya $\hat{\beta}$ memiliki variansi terkecil di antara semua penduga linear tak bias.

Bukti. Penduga OLS dapat dituliskan sebagai

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \quad (2.102)$$

Substitusi $\mathbf{y} = \mathbf{X}\beta + \varepsilon$ menghasilkan

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top (\mathbf{X}\beta + \varepsilon) = \beta + (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \varepsilon, \quad (2.103)$$

sehingga rata-rata dari $\hat{\beta}$ adalah

$$\mathbb{E}[\hat{\beta}] = \beta + (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbb{E}[\varepsilon] = \beta, \quad (2.104)$$

yang menunjukkan bahwa $\hat{\beta}$ tak bias. Kovariansnya diperoleh dari

$$\text{Var}(\hat{\beta}) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \text{Var}(\varepsilon) \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}. \quad (2.105)$$

Untuk sifat BLUE, dipertimbangkan penduga linear tak bias umum $\tilde{\beta} = \mathbf{C}\mathbf{y}$ dengan $\mathbf{C}\mathbf{X} = \mathbf{I}$. Variansi dari $\tilde{\beta}$ adalah

$$\text{Var}(\tilde{\beta}) = \mathbf{C}\sigma^2\mathbf{I}\mathbf{C}^\top = \sigma^2\mathbf{C}\mathbf{C}^\top. \quad (2.106)$$

Melalui dekomposisi $\mathbf{C} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top + \mathbf{D}$ dengan $\mathbf{D}\mathbf{X} = \mathbf{0}$, diperoleh

$$\text{Var}(\tilde{\beta}) = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1} + \sigma^2 \mathbf{D}\mathbf{D}^\top \succeq \text{Var}(\hat{\beta}), \quad (2.107)$$

yang membuktikan bahwa $\hat{\beta}$ memiliki variansi minimum di antara semua penduga linear tak bias. ■

Teorema 2.2.5 (Kepadanan OLS dan MLE, (Montgomery dkk., 2012)) *Jika asumsi normalitas galat terpenuhi, maka penduga kemungkinan maksimum (PKM) atau maximum likelihood estimation (MLE) untuk β identik dengan penduga kua-*

drat terkecil (OLS), yaitu

$$\hat{\beta}_{MLE} = \hat{\beta}_{OLS} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \quad (2.108)$$

Penduga MLE untuk varians galat adalah

$$\hat{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{X} \hat{\beta})^\top (\mathbf{y} - \mathbf{X} \hat{\beta}). \quad (2.109)$$

Bukti. Dengan asumsi $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, fungsi *likelihood* untuk β dan σ^2 adalah

$$L(\beta, \sigma^2 \mid \mathbf{y}) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta)\right), \quad (2.110)$$

yang memiliki log-*likelihood*

$$\ell(\beta, \sigma^2) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta). \quad (2.111)$$

Maksimisasi terhadap β dengan turunan pertama didapatkan

$$\begin{aligned} \frac{\partial \ell(\beta, \sigma^2)}{\partial \beta} &= \frac{2\mathbf{X}^\top}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta) \\ &= \frac{1}{\sigma^2} \mathbf{X}^\top \mathbf{y} - \frac{1}{\sigma^2} \mathbf{X}^\top \mathbf{X} \beta \\ \mathbf{0} &= \mathbf{X}^\top \mathbf{y} - \mathbf{X}^\top \mathbf{X} \hat{\beta} \\ \mathbf{X}^\top \mathbf{X} \hat{\beta} &= \mathbf{X}^\top \mathbf{y} \\ \hat{\beta} &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \end{aligned} \quad (2.112)$$

Perhatikan bahwa turunan kedua terhadap β menghasilkan $-\frac{1}{\sigma^2} (\mathbf{X}^\top \mathbf{X})$. Sebab $\mathbf{X}^\top \mathbf{X} \succeq 0$, maka $\frac{\partial \ell}{\partial \beta} \preceq 0$ yang menjadikan $\hat{\beta}$ memaksimumkan fungsi log-*likelihood*. Oleh karena itu

$$\hat{\beta}_{MLE} = \hat{\beta}_{OLS}. \quad (2.113)$$

Selanjutnya, turunan pertama ℓ terhadap σ^2 akan menghasilkan

$$\begin{aligned}\frac{\partial \ell(\boldsymbol{\beta}, \sigma^2)}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ 0 &= -n\hat{\sigma}^2 + (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\ \hat{\sigma}^2 &= \frac{1}{n}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}).\end{aligned}\tag{2.114}$$

■

2.2.2. Regresi Terboboti

Regresi terboboti dibentuk apabila model regresi linear klasik melanggar asumsi homoskedastisitas galat. Heteroskedastisitas terjadi apabila komponen galat regresi tidak memiliki variansi yang konstan. Apabila terjadi heteroskedastisitas, maka Teorema Gauss–Markov tidak berlaku sehingga penduga dari model regresi tidak lagi bersifat BLUE.

Definisi 2.2.6 (Model Regresi Linear dengan Heteroskedastisitas, (Greene, 2002; Gujarati & Porter, 2009; Montgomery dkk., 2012)) *Pertimbangkan model regresi linear*

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}, \tag{2.115}$$

dengan struktur ragam–kovarians galat

$$\text{Var}(\boldsymbol{\varepsilon} \mid \mathbf{X}) = \sigma^2 \boldsymbol{\Omega}, \tag{2.116}$$

dengan $\boldsymbol{\Omega}$ adalah matriks simetri definit-positif berukuran $n \times n$. Model yang memperkenalkan struktur variansi-kovarians galat disebut sebagai model kuadrat terkecil tergeneralisasi atau generalized least squares (GLS).

Kasus umum dari GLS yang sering dipakai adalah kuadrat terkecil terboboti atau *weighted least squares* (WLS), yaitu $\boldsymbol{\Omega} = \text{diag}(\omega_1, \dots, \omega_n)$ yang merupakan heteroskedastisitas diagonal dengan bobot $\mathbf{W} = \boldsymbol{\Omega}^{-1} = \text{diag}(w_1, \dots, w_n)$, $w_i = 1/\omega_i$.

Penduga GLS diperoleh dengan meminimumkan jumlah kuadrat terbobot

$$S_{\mathbf{W}}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\top} \mathbf{W} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \quad (2.117)$$

dengan $\mathbf{W} = \boldsymbol{\Omega}^{-1}$. Diferensiasi terhadap $\boldsymbol{\beta}$ dan disamakan dengan nol memberikan

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\beta}} S_{\mathbf{W}}(\boldsymbol{\beta}) &= -2\mathbf{W}\mathbf{X}^{\top}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \quad (\text{karena } \mathbf{W} \text{ simetris}) \\ 0 &= -2\mathbf{X}^{\top}\mathbf{W}\mathbf{y} - 2\mathbf{X}^{\top}\mathbf{W}\mathbf{X}\hat{\boldsymbol{\beta}} \\ \mathbf{X}^{\top}\mathbf{W}\mathbf{X}\hat{\boldsymbol{\beta}} &= \mathbf{X}^{\top}\mathbf{W}\mathbf{y}, \end{aligned} \quad (2.118)$$

sehingga pada asumsi $\mathbf{X}^{\top}\mathbf{W}\mathbf{X}$ nonsingular,

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = (\mathbf{X}^{\top}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{W}\mathbf{y}. \quad (2.119)$$

Penduga ini tak bias jika \mathbf{W} tidak acak atau independen dari $\boldsymbol{\varepsilon}$ dan $\mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$, sebab $\mathbb{E}[\hat{\boldsymbol{\beta}}_{\text{GLS}} \mid \mathbf{X}] = \boldsymbol{\beta}$.

Teorema 2.2.7 (Teorema Aitken atau Generalisasi Gauss–Markov, (Greene, 2002; Montgomery dkk., 2012)) Misalkan model regresi linear

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}, \quad \text{Var}(\boldsymbol{\varepsilon} \mid \mathbf{X}) = \sigma^2\boldsymbol{\Omega},$$

dengan $\boldsymbol{\Omega}$ simetris definit-positif, maka penduga

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = (\mathbf{X}^{\top}\boldsymbol{\Omega}^{-1}\mathbf{X})^{-1}\mathbf{X}^{\top}\boldsymbol{\Omega}^{-1}\mathbf{y} \quad (2.120)$$

adalah best linear unbiased estimator (BLUE) dari $\boldsymbol{\beta}$.

Bukti. Pertama-tama perhatikan bahwa setiap penduga linear bagi $\boldsymbol{\beta}$ dapat ditulis dalam bentuk $\tilde{\boldsymbol{\beta}} = \mathbf{A}\mathbf{y}$ dengan $\mathbf{A} \in \mathbb{R}^{p \times n}$. Agar tak bias, berlaku

$$\mathbb{E}[\tilde{\boldsymbol{\beta}} \mid \mathbf{X}] = \mathbf{A}\mathbb{E}[\mathbf{y} \mid \mathbf{X}] = \mathbf{A}\mathbf{X}\boldsymbol{\beta}. \quad (2.121)$$

Supaya sama dengan $\boldsymbol{\beta}$ untuk semua $\boldsymbol{\beta}$, syarat yang harus dipenuhi adalah $\mathbf{A}\mathbf{X} =$

\mathbf{I}_p . Selanjutnya, kovarians dari $\tilde{\beta}$ diberikan oleh

$$\text{Var}(\tilde{\beta} \mid \mathbf{X}) = \mathbf{A} \sigma^2 \boldsymbol{\Omega} \mathbf{A}^\top. \quad (2.122)$$

Di antara semua matriks \mathbf{A} yang memenuhi $\mathbf{A}\mathbf{X} = \mathbf{I}_p$, pilih

$$\mathbf{A}_0 = (\mathbf{X}^\top \boldsymbol{\Omega}^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\Omega}^{-1}. \quad (2.123)$$

Jelas bahwa $\mathbf{A}_0 \mathbf{X} = \mathbf{I}_p$, sehingga \mathbf{A}_0 menghasilkan penduga tak bias. Penduga inilah yang dikenal sebagai *generalized least squares*.

Sekarang, ambil sembarang penduga tak bias lain \mathbf{A} . Matriks \mathbf{A} tersebut dapat ditulis sebagai $\mathbf{A} = \mathbf{A}_0 + \mathbf{D}$ dengan $\mathbf{D}\mathbf{X} = \mathbf{0}$, sebab $\mathbf{A}_0 \mathbf{X} = \mathbf{I}_p$ dan syarat $\mathbf{A}\mathbf{X} = \mathbf{I}_p$ harus tetap dipenuhi. Dengan demikian,

$$\tilde{\beta} = \mathbf{A}\mathbf{y} = \mathbf{A}_0 \mathbf{y} + \mathbf{D}\mathbf{y}. \quad (2.124)$$

Variansi dari penduga ini adalah

$$\text{Var}(\tilde{\beta} \mid \mathbf{X}) = \text{Var}(\mathbf{A}_0 \mathbf{y} \mid \mathbf{X}) + \sigma^2 \mathbf{D} \boldsymbol{\Omega} \mathbf{D}^\top. \quad (2.125)$$

Bagian pertama adalah kovarians dari $\hat{\beta}_{\text{GLS}}$, sedangkan bagian kedua selalu semi definit positif karena $\boldsymbol{\Omega}$ definit-positif. Hal ini berarti tambahan $\mathbf{D}\mathbf{y}$ hanya menambah variansi tanpa mengurangi bias. Dengan demikian, kovarians dari sembarang penduga linear tak bias selalu lebih besar atau sama dengan kovarians GLS. Oleh karena itu, $\hat{\beta}_{\text{GLS}}$ meminimalkan kovarians di antara semua penduga linear tak bias, sehingga merupakan BLUE. ■

Teorema 2.2.8 (Kepadanan GLS dan MLE di Bawah Normalitas, (Greene, 2002; Montgomery dkk., 2012)) Jika $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{\Omega})$, maka penduga maksimum likelihood (MLE) untuk β adalah

$$\hat{\beta}_{\text{MLE}} = \hat{\beta}_{\text{GLS}} = (\mathbf{X}^\top \boldsymbol{\Omega}^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\Omega}^{-1} \mathbf{y}, \quad (2.126)$$

dan MLE untuk σ^2 (dengan Ω diketahui) adalah

$$\hat{\sigma}_{MLE}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{X}\hat{\beta})^\top \Omega^{-1} (\mathbf{y} - \mathbf{X}\hat{\beta}). \quad (2.127)$$

Bukti. Di bawah normalitas galat, fungsi *likelihood* untuk β dan σ^2 adalah

$$L(\beta, \sigma^2 | \mathbf{y}) = (2\pi\sigma^2)^{-n/2} |\Omega|^{-1/2} \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top \Omega^{-1} (\mathbf{y} - \mathbf{X}\beta)\right).$$

Hal ini berarti fungsi log-*likelihood* untuk β dan σ^2 adalah

$$\ell(\beta, \sigma^2) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2} \log |\Omega| - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top \Omega^{-1} (\mathbf{y} - \mathbf{X}\beta).$$

Maksimisasi ℓ untuk β dengan turunan pertama menghasilkan

$$\begin{aligned} \frac{\partial \ell(\beta, \sigma^2 | \mathbf{y})}{\partial \beta} &= \frac{2}{2\sigma^2} \mathbf{X}^\top \Omega^{-1} (\mathbf{y} - \mathbf{X}\beta) \\ 0 &= \frac{1}{\sigma^2} \mathbf{X}^\top \Omega^{-1} \mathbf{y} - \frac{1}{\sigma^2} \mathbf{X}^\top \Omega^{-1} \mathbf{X}\hat{\beta} \\ \frac{1}{\sigma^2} \mathbf{X}^\top \Omega^{-1} \mathbf{X}\hat{\beta} &= \frac{1}{\sigma^2} \mathbf{X}^\top \Omega^{-1} \mathbf{y} \\ \hat{\beta} &= (\mathbf{X}^\top \Omega^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \Omega^{-1} \mathbf{y}, \end{aligned} \quad (2.128)$$

sehingga $\hat{\beta}_{MLE} = \hat{\beta}_{GLS}$. Selanjutnya, turunan terhadap σ^2 memberi $\partial \ell / \partial \sigma^2 = -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} (\mathbf{y} - \mathbf{X}\hat{\beta})^\top \Omega^{-1} (\mathbf{y} - \mathbf{X}\hat{\beta}) = 0$, yang menghasilkan rumus $\hat{\sigma}_{MLE}^2$ di atas. ■

Dalam analisis nyata, Ω biasanya jarang diketahui. *Feasible GLS* (FGLS) memperkirakan Ω , misalnya dari residu OLS atau model varian parametrik, lalu menyusun $\widehat{\mathbf{W}} = \widehat{\Omega}^{-1}$ dan menghitung $\hat{\beta}_{FGLS} = (\mathbf{X}^\top \widehat{\mathbf{W}} \mathbf{X})^{-1} \mathbf{X}^\top \widehat{\mathbf{W}} \mathbf{y}$. Di bawah kondisi reguler, $\hat{\beta}_{FGLS}$ konsisten dan asimtotik efisien. Namun, untuk sampel kecil, sifat *finite-sample* bergantung pada kualitas spesifikasi $\widehat{\Omega}$.

2.2.3. Model Koefisien Bervariasi

Model koefisien bervariasi merupakan perluasan dari model regresi linear klasik yang memperbolehkan koefisien regresi berubah secara sistematis sesuai dengan karakteristik tertentu, seperti lokasi geografis, waktu, atau variabel lainnya. Model ini merupakan jawaban terhadap keterbatasan model regresi tradisional yang mengasumsikan hubungan konstan antara variabel independen dan dependen di seluruh wilayah observasi. (Fan & Zhang, 2008)

Definisi 2.2.9 (Model Koefisien Bervariasi Umum, (Fan & Zhang, 2008)) *Model koefisien bervariasi dapat diformulasikan sebagai*

$$Y_i = \sum_{k=0}^p \beta_k(z_i) X_{ik} + \varepsilon_i, \quad i = 1, 2, \dots, n, \quad (2.129)$$

dengan komponen-komponen sebagai berikut.

- (a) Y_i merupakan variabel dependen pada observasi ke- i .
- (b) X_{ik} merupakan variabel independen ke- k pada observasi ke- i , dengan $X_{i0} = 1$.
- (c) $\beta_k(z_i)$ merupakan koefisien yang bervariasi untuk variabel ke- k sebagai fungsi dari vektor karakteristik z_i .
- (d) z_i merupakan vektor variabel yang menentukan variasi koefisien.
- (e) ε_i merupakan galat acak dengan $\mathbb{E}[\varepsilon_i] = 0$ dan $\text{Var}(\varepsilon_i) = \sigma^2$.

Pendekatan utama untuk mengestimasi model koefisien bervariasi adalah melalui regresi linear lokal yang memberikan fleksibilitas dalam menangkap variasi koefisien tanpa perlu menspesifikasi bentuk fungsional secara parametrik.

Definisi 2.2.10 (Regresi Linear Lokal untuk Model Koefisien Bervariasi, (Fan & Zhang, 2008)) *Untuk setiap titik z_0 , koefisien $\beta_k(z_0)$ diestimasi dengan me-*

minimumkan jumlah kuadrat tertimbang lokal, yaitu

$$\sum_{i=1}^n K_h(z_i - z_0) \left[Y_i - \sum_{k=0}^p \beta_k X_{ik} \right]^2, \quad (2.130)$$

dengan komponen-komponen

(a) $K_h(\cdot)$ adalah fungsi kernel dengan bandwidth h dan

(b) $K_h(u) = K(u/h)/h$ untuk kernel univariat atau $K_h(u) = K(u/h)/h^d$ untuk kernel multivariat dimensi d .

Fungsi kernel umum yang digunakan adalah Gaussian, yaitu $K(u) = (2\pi)^{-1/2} \exp(-u^2/2)$.

Teorema 2.2.11 (Estimator Regresi Linear Lokal) *Estimator regresi linear lokal untuk $\beta_k(z_0)$ diberikan oleh:*

$$\hat{\beta}(z_0) = (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}(z_0) \mathbf{y}, \quad (2.131)$$

dengan $\mathbf{W}(z_0) = \text{diag}\{K_h(z_1 - z_0), \dots, K_h(z_n - z_0)\}$ adalah matriks bobot diagonal.

Teorema 2.2.12 (Pemilihan Bandwidth Optimal, (Fan & Zhang, 2008)) *bandwidth optimal h dapat dipilih menggunakan metode cross-validation dengan meminimumkan*

$$\text{CV}(h) = \frac{1}{n} \sum_{i=1}^n \left[Y_i - \hat{Y}_{-i}(X_i, z_i; h) \right]^2, \quad (2.132)$$

dengan $\hat{Y}_{-i}(X_i, z_i; h)$ adalah prediksi leave-one-out untuk observasi ke- i menggunakan bandwidth h .

Teorema 2.2.13 (Sifat Asimtotik Estimator) *Di bawah kondisi reguler, estimator regresi linear lokal memiliki sifat:*

(a) *bias, yaitu $\mathbb{E}[\hat{\beta}_k(z_0)] - \beta_k(z_0) = \mathcal{O}(h^2)$ jika $\beta_k(\cdot)$ memiliki turunan kedua yang kontinu;*

(b) $\text{Var}(\hat{\beta}_k(z_0)) = \mathcal{O}((nh)^{-1})$; dan

(c) bandwidth optimal yang meminimumkan rata-rata kuadrat galat (MSE) proporsional dengan $n^{-1/5}$.

Teorema 2.2.14 (Interval Konfidensi untuk Model Koefisien Bervariasi, (Fan & Zhang, 2008)) Interval konfidensi titik demi titik untuk $\beta_k(z_0)$ dapat dikonstruksikan sebagai

$$\hat{\beta}_k(z_0) \pm z_{\alpha/2} \sqrt{\hat{\sigma}^2 \mathbf{e}_k^\top (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{e}_k}, \quad (2.133)$$

dengan \mathbf{e}_k adalah vektor unit ke- k dan $\hat{\sigma}^2$ adalah estimator variansi residual.

Bukti. Misalkan model koefisien bervariasi $Y_i = \mathbf{x}_i^\top \boldsymbol{\beta}(z_i) + \varepsilon_i$ dengan $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ yang bersifat *i.i.d.* Estimator regresi linear lokal diberikan oleh

$$\hat{\boldsymbol{\beta}}(z_0) = (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}(z_0) \mathbf{y} \quad (2.134)$$

dengan $\mathbf{W}(z_0) = \text{diag}\{K_h(z_1 - z_0), \dots, K_h(z_n - z_0)\}$. Di bawah asumsi regularitas *kernel* dan $n \rightarrow \infty$, $nh \rightarrow \infty$, bias estimator adalah $\mathcal{O}(h^2)$ sehingga dapat diabaikan secara asimtotik. Oleh karena itu

$$\hat{\boldsymbol{\beta}}(z_0) \approx \mathcal{N}(\boldsymbol{\beta}(z_0), \sigma^2 (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1}). \quad (2.135)$$

Untuk komponen ke- k , dengan \mathbf{e}_k sebagai vektor unit ke- k berlaku

$$\hat{\beta}_k(z_0) = \mathbf{e}_k^\top \hat{\boldsymbol{\beta}}(z_0), \quad (2.136)$$

$$\mathbb{E}[\hat{\beta}_k(z_0)] = \mathbf{e}_k^\top \boldsymbol{\beta}(z_0) = \beta_k(z_0), \text{ dan} \quad (2.137)$$

$$\begin{aligned} \text{Var}(\hat{\beta}_k(z_0)) &= \mathbf{e}_k^\top [\sigma^2 (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1}] \mathbf{e}_k. \\ &= \sigma^2 \mathbf{e}_k^\top (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{e}_k. \end{aligned} \quad (2.138)$$

Selanjutnya, karena $\hat{\beta}_k(z_0) \sim \mathcal{N}(\beta_k(z_0), \sigma^2 \mathbf{e}_k^\top (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{e}_k)$, maka

$$Z = \frac{\hat{\beta}_k(z_0) - \beta_k(z_0)}{\sigma \sqrt{\mathbf{e}_k^\top (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{e}_k}} \sim \mathcal{N}(0, 1) \quad (2.139)$$

Sebab σ^2 tidak diketahui, digunakan estimator konsisten, yaitu

$$\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^n (Y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}(z_i))^2. \quad (2.140)$$

Di bawah normalitas error, $\hat{\sigma}^2$ independen dari $\hat{\boldsymbol{\beta}}(z_0)$ secara asimtotik.

Dengan mengganti σ dengan $\hat{\sigma}$, diperoleh

$$\hat{Z} = \frac{\hat{\beta}_k(z_0) - \beta_k(z_0)}{\hat{\sigma} \sqrt{\mathbf{e}_k^\top (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{e}_k}} \xrightarrow{d} \mathcal{N}(0, 1). \quad (2.141)$$

Oleh karena itu, untuk tingkat konfidensi $(1 - \alpha) \times 100\%$ didapatkan

$$P\left(-z_{\alpha/2} \leq \hat{Z} \leq z_{\alpha/2}\right) = 1 - \alpha, \quad (2.142)$$

sehingga memberikan interval konfidensi

$$\hat{\beta}_k(z_0) \pm z_{\alpha/2} \sqrt{\hat{\sigma}^2 \mathbf{e}_k^\top (\mathbf{X}^\top \mathbf{W}(z_0) \mathbf{X})^{-1} \mathbf{e}_k} \quad (2.143)$$

■

Validasi model koefisien bervariasi dapat dilakukan melalui berbagai teknik, termasuk *cross-validation*, *bootstrap*, dan analisis residual. Pemilihan *bandwidth* yang optimal merupakan aspek kritis yang menentukan keseimbangan antara bias dan variansi estimator.

Contoh 2.2.15 Untuk model dengan satu kovariat yang bervariasi terhadap lokasi z , seperti

$$Y_i = \beta_0(z_i) + \beta_1(z_i)X_i + \varepsilon_i, \quad (2.144)$$

estimasi dilakukan dengan meminimumkan

$$\sum_{j=1}^n K_h(z_j - z_i) [Y_j - \beta_0 - \beta_1 X_j]^2 \quad (2.145)$$

untuk setiap titik z_i . Hal ini menghasilkan estimator lokal yang adaptif terhadap

karakteristik data pada setiap lokasi.

Model koefisien bervariasi menyediakan kerangka kerja yang fleksibel untuk menganalisis heterogenitas spasial dan temporal dalam hubungan statistik, dengan aplikasi yang luas dalam ekonometrika spasial, epidemiologi, dan analisis lingkungan.

2.2.4. Regresi dengan Dependensi Spasial dan Spasial Temporal

Dependensi spasial (*spatial dependence*) atau autokorelasi spasial terjadi ketika nilai suatu variabel di suatu lokasi berkorelasi dengan nilai variabel yang sama di lokasi lain yang berdekatan secara geografis. Konsep ini melanggar asumsi independensi dalam regresi linear klasik dan memerlukan perlakuan khusus dalam pemodelan ekonometrika spasial (Anselin, 1988; LeSage & Pace, 2009).

Definisi 2.2.16 (Dependensi Spasial) Misalkan Y_i merupakan nilai variabel di lokasi i dan Y_j merupakan nilai variabel di lokasi tetangga j . Dependensi spasial terjadi jika

$$\text{Cov}(Y_i, Y_j) \neq 0, \quad \text{untuk lokasi } i \neq j \text{ yang berdekatan}, \quad (2.146)$$

yang dapat diukur menggunakan statistik Moran's I , yaitu

$$I = \frac{n}{\sum_{i=1}^n \sum_{j=1}^n w_{ij}} \cdot \frac{\sum_{i=1}^n \sum_{j=1}^n w_{ij} (Y_i - \bar{Y})(Y_j - \bar{Y})}{\sum_{i=1}^n (Y_i - \bar{Y})^2}, \quad (2.147)$$

dengan w_{ij} adalah bobot spasial antara lokasi i dan j .

Model regresi dengan dependensi spasial dapat diklasifikasikan menjadi beberapa bentuk utama berdasarkan letak dependensi spasial muncul dalam spesifikasi model.

Definisi 2.2.17 (Spatial Autoregressive Model (SAR)) Model SAR atau spasial lag model mengasumsikan bahwa variabel dependen di suatu lokasi dipengaruhi oleh

nilai variabel dependen di lokasi tetangga, yaitu

$$y_i = \rho \sum_{j=1}^n w_{ij} y_j + \sum_{k=1}^p \beta_k x_{ik} + \varepsilon_i, \quad (2.148)$$

atau dalam bentuk matriksnya adalah

$$\mathbf{y} = \rho \mathbf{W} \mathbf{y} + \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (2.149)$$

dengan ρ adalah parameter autokorelasi spasial dan \mathbf{W} adalah matriks bobot spasial.

Definisi 2.2.18 (Spatial Error Model (SEM)) Model SEM mengasumsikan bahwa dependensi spasial terjadi pada komponen galat, yaitu

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \mathbf{u}, \quad \mathbf{u} = \lambda \mathbf{W} \mathbf{u} + \boldsymbol{\varepsilon}, \quad (2.150)$$

dengan λ adalah parameter autokorelasi spasial pada galat dan $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$.

Definisi 2.2.19 (Spatial Durbin Model (SDM)) Model SDM menggabungkan lag spasial dari variabel dependen dan independen, yaitu

$$\mathbf{y} = \rho \mathbf{W} \mathbf{y} + \mathbf{X} \boldsymbol{\beta} + \mathbf{W} \mathbf{X} \boldsymbol{\delta} + \boldsymbol{\varepsilon}, \quad (2.151)$$

dengan $\boldsymbol{\delta}$ adalah vektor parameter untuk lag spasial variabel independen.

Untuk kasus spasial temporal, model-model di atas dapat diperluas untuk mengakomodasi dimensi waktu.

Definisi 2.2.20 (Space-Time Autoregressive (STAR), (Amri, 2020)) Model STAR merupakan gabungan model autoregresif waktu dan spasial dengan asumsi parameter yang homogen di seluruh lokasi. Bentuk umum model $STAR(p; \lambda_k)$ adalah

$$\mathbf{z}_t = \sum_{k=1}^p \sum_{\ell=0}^{\lambda_k} \Phi_k \mathbf{W}^{(\ell)} \mathbf{z}_{t-k} + \boldsymbol{\varepsilon}_t, \quad (2.152)$$

dengan Φ_k adalah matriks parameter waktu orde k , $\mathbf{W}^{(\ell)}$ adalah matriks bobot spasial orde ℓ , dan ε_t adalah vektor galat. Model ini mengasumsikan karakteristik lokasi homogen.

Definisi 2.2.21 (Generalized Space-Time Autoregressive (GSTAR), (Amri, 2020))

Model GSTAR merupakan generalisasi dari STAR yang mengizinkan parameter berbeda untuk setiap lokasi (heterogen). Bentuk umum model $\text{GSTAR}(p; \lambda_k)$ adalah

$$\mathbf{y}_t = \sum_{k=1}^p \sum_{\ell=0}^{\lambda_k} \Phi_k \mathbf{W}^{(\ell)} \mathbf{y}_{t-k} + \varepsilon_t, \quad (2.153)$$

dengan $\mathbf{y}_t = \mathbf{z}_t - \mathbf{z}_{t-1}$ dan $\mathbf{y}_{t-k} = \mathbf{z}_{t-k} - \mathbf{z}_{t-k-1}$. Model ini memungkinkan variasi parameter spasial di seluruh lokasi.

Model-model ini menyediakan kerangka yang fleksibel untuk menangkap dependensi baik dalam dimensi spasial maupun temporal. Pemilihan model bergantung pada karakteristik data dan hipotesis yang ingin diuji. Model STAR cocok untuk kasus dengan parameter homogen antar lokasi, sedangkan GSTAR lebih sesuai ketika terdapat heterogenitas spasial dalam struktur autoregresif.

2.2.5. Regresi dengan Heterogenitas Spasial dan Spasial Temporal

Heterogenitas spasial (*spatial heterogeneity*) terjadi ketika hubungan statistik antara variabel berbeda di berbagai lokasi geografis. Berbeda dengan dependensi spasial yang menangkap korelasi antar lokasi, heterogenitas spasial menangkap variasi dalam struktur hubungan itu sendiri (Fotheringham dkk., 2002).

Definisi 2.2.22 (Heterogenitas Spasial) Heterogenitas spasial terjadi ketika parameter regresi β bervariasi menurut lokasi geografis, yaitu

$$\beta = \beta(s_i) = \beta(u_i, v_i), \quad (2.154)$$

dengan (u_i, v_i) adalah koordinat geografis lokasi ke- i . Hal ini mengindikasikan bahwa hubungan struktural yang sama dapat memiliki kekuatan dan arah yang

berbeda di lokasi yang berbeda.

Model utama untuk menangkap heterogenitas spasial adalah *Geographically Weighted Regression* (GWR) yang memungkinkan koefisien regresi bervariasi secara kontinu di ruang geografis.

Definisi 2.2.23 (Geographically Weighted Regression (GWR)) Model GWR memungkinkan koefisien regresi bervariasi menurut lokasi:

$$y_i = \beta_0(u_i, v_i) + \sum_{k=1}^p \beta_k(u_i, v_i) x_{ik} + \varepsilon_i, \quad (2.155)$$

dengan $\beta_k(u_i, v_i)$ adalah koefisien regresi untuk variabel k di lokasi (u_i, v_i) .

Estimasi parameter GWR dilakukan menggunakan *locally weighted least squares* (LWLS), yang memberikan bobot lebih besar pada observasi yang berdekatan secara geografis.

Teorema 2.2.24 (Estimator LWLS untuk GWR) Untuk setiap lokasi i , koefisien lokal diestimasi dengan:

$$\hat{\beta}(u_i, v_i) = (\mathbf{X}^\top \mathbf{W}_i \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_i \mathbf{y}, \quad (2.156)$$

dengan \mathbf{W}_i adalah matriks diagonal bobot dengan elemen

$$w_{ij} = \exp \left(-\frac{d_{ij}^2}{h^2} \right), \quad (2.157)$$

dengan d_{ij} adalah jarak antara lokasi i dan j , dan h adalah bandwidth.

Bukti. Pertama-tama didefinisikan masalah optimisasi lokal. Untuk setiap lokasi (u_i, v_i) , akan diminimumkan jumlah kuadrat residual tertimbang lokal. Misalkan model GWR diberikan oleh $y_j = \mathbf{x}_j^\top \boldsymbol{\beta}(u_i, v_i) + \varepsilon_j$ untuk observasi $j = 1, 2, \dots, n$.

Fungsi objektif yang akan diminimumkan adalah

$$S_i(\boldsymbol{\beta}) = \sum_{j=1}^n w_{ij} (y_j - \mathbf{x}_j^\top \boldsymbol{\beta})^2, \quad (2.158)$$

dengan w_{ij} adalah bobot yang mencerminkan kedekatan geografis antara lokasi i dan j .

Untuk mencari nilai β yang meminimumkan $S_i(\beta)$, diambil turunan pertama terhadap β dan disamakan dengan nol. Dapat menulis ulang $S_i(\beta)$ dalam bentuk matriks sebagai

$$S_i(\beta) = (\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{W}_i (\mathbf{y} - \mathbf{X}\beta), \quad (2.159)$$

dengan $\mathbf{W}_i = \text{diag}(w_{i1}, w_{i2}, \dots, w_{in})$ adalah matriks diagonal berukuran $n \times n$. Turunan pertama dari $S_i(\beta)$ terhadap β adalah

$$\frac{\partial S_i(\beta)}{\partial \beta} = \frac{\partial}{\partial \beta} [(\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{W}_i (\mathbf{y} - \mathbf{X}\beta)] \quad (2.160)$$

$$= -2\mathbf{X}^\top \mathbf{W}_i (\mathbf{y} - \mathbf{X}\beta) \quad (2.161)$$

Selanjutnya, dengan menyamakan turunan pertama dengan nol, diperoleh

$$-2\mathbf{X}^\top \mathbf{W}_i (\mathbf{y} - \mathbf{X}\hat{\beta}) = 0, \quad (2.162)$$

sehingga

$$\mathbf{X}^\top \mathbf{W}_i \mathbf{y} - \mathbf{X}^\top \mathbf{W}_i \mathbf{X} \hat{\beta} = 0 \quad (2.163)$$

$$\mathbf{X}^\top \mathbf{W}_i \mathbf{X} \hat{\beta} = \mathbf{X}^\top \mathbf{W}_i \mathbf{y} \quad (2.164)$$

Dengan asumsi bahwa matriks $\mathbf{X}^\top \mathbf{W}_i \mathbf{X}$ adalah non-singular (invertibel), kita dapat menyelesaikan untuk $\hat{\beta}$:

$$\hat{\beta}(u_i, v_i) = (\mathbf{X}^\top \mathbf{W}_i \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_i \mathbf{y} \quad (2.165)$$

■

Perlu diperhatikan bahwa bobot dalam GWR merupakan bobot diagonal yang berarti bahwa model ini merupakan perluasan dari model regresi linear terboboti (WLS) yang telah dibahas sebelumnya.

Definisi 2.2.25 (Fungsi kernel untuk Pembobotan Spasial) Beberapa fungsi kernel yang umum digunakan dalam GWR adalah sebagai berikut.

- (a) Kernel Gaussian, yaitu $w_{ij} = \exp \left(-\frac{1}{2} \left(\frac{d_{ij}}{h} \right)^2 \right)$.
- (b) Kernel Bi-square, yaitu $w_{ij} = \begin{cases} \left(1 - \left(\frac{d_{ij}}{h} \right)^2 \right)^2 & \text{jika } d_{ij} < h \\ 0 & \text{jika } d_{ij} \geq h. \end{cases}$
- (c) Kernel Tri-cube, yaitu $w_{ij} = \begin{cases} \left(1 - \left(\frac{d_{ij}}{h} \right)^3 \right)^3 & \text{jika } d_{ij} < h \\ 0 & \text{jika } d_{ij} \geq h. \end{cases}$

Pemilihan *bandwidth* h sangat krusial dalam GWR dan dapat dilakukan melalui *cross-validation* atau validasi silang.

Definisi 2.2.26 (Cross-Validation untuk Bandwidth) Bandwidth optimal h^* dipilih dengan meminimalkan fungsi cross-validation:

$$CV(h) = \sum_{i=1}^n (y_i - \hat{y}_{\neq i}(h))^2, \quad (2.166)$$

dengan $\hat{y}_{\neq i}(h)$ adalah prediksi untuk observasi i menggunakan model yang di-fit tanpa observasi i . Hal ini berarti bahwa

$$h^* = \arg \min_h CV(h). \quad (2.167)$$

Untuk data panel spasial, GWR dapat diperluas menjadi *Geographically and Temporally Weighted Regression* (GTWR).

Definisi 2.2.27 (Geographically and Temporally Weighted Regression (GTWR)) Model GTWR memungkinkan koefisien bervariasi baik secara spasial maupun temporal

$$y_{it} = \beta_0(u_i, v_i, t) + \sum_{k=1}^p \beta_k(u_i, v_i, t)x_{ikt} + \varepsilon_{it} \quad (2.168)$$

dengan estimator

$$\hat{\beta}(u_i, v_i, t) = (\mathbf{X}^\top \mathbf{W}_{i,t} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_{i,t} \mathbf{y}, \quad (2.169)$$

dengan $\mathbf{W}_{i,t}$ menggabungkan bobot spasial dan temporal, yaitu

$$w_{ij,t} = \exp \left(-\frac{d_{ij}^2}{h_s^2} - \frac{(t_i - t_j)^2}{h_t^2} \right) \quad (2.170)$$

dengan h_s dan h_t adalah bandwidth spasial dan temporal secara berturut-turut.

Teorema 2.2.28 (Uji Heterogenitas Spasial) Untuk menguji keberadaan heterogenitas spasial, dapat digunakan uji Breusch-Pagan lokal:

$$\text{BP} = \frac{1}{2} \mathbf{f}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{f}, \quad (2.171)$$

dengan $\mathbf{f} = \mathbf{X}^\top \mathbf{A} \mathbf{e}^2$ dengan \mathbf{A} adalah matriks transformasi spasial dan \mathbf{e}^2 adalah vektor residual kuadrat. Di bawah hipotesis nol bahwa terdapat homoskedastisitas spasial, statistik BP mengikuti distribusi χ^2 dengan derajat bebas p .

Bukti. Dimulai dengan model regresi linear $\mathbf{y} = \mathbf{X}\beta + \varepsilon$ untuk menguji apakah terdapat heterogenitas spasial. Hipotesis yang akan diuji adalah

$$H_0 : \text{Var}(\varepsilon_i) = \sigma^2 \text{ untuk semua } i \text{ (homoskedastisitas)}, \quad (2.172)$$

$$H_1 : \text{Var}(\varepsilon_i) = \sigma^2 h(\mathbf{z}_i^\top \boldsymbol{\alpha}) \text{ (heteroskedastisitas spasial)}, \quad (2.173)$$

dengan \mathbf{z}_i adalah vektor yang berisi karakteristik spasial lokasi i , dan $h(\cdot)$ adalah fungsi yang menghubungkan karakteristik spasial dengan variansi galat. Langkah pertama adalah menentukan bentuk fungsi $h(\cdot)$. Dipilih $h(t) = \exp(t)$ sehingga variansi galat dapat dituliskan sebagai $\text{Var}(\varepsilon_i) = \sigma^2 \exp(\mathbf{z}_i^\top \boldsymbol{\alpha})$. Dengan mengambil logaritma, didapatkan

$$\log(\text{Var}(\varepsilon_i)) = \log(\sigma^2) + \mathbf{z}_i^\top \boldsymbol{\alpha}.$$

Di bawah hipotesis nol H_0 , parameter $\alpha = \mathbf{0}$, yang berarti tidak ada heterogenitas spasial. Dalam kasus ini, nilai harapan dari kuadrat galat adalah

$$\mathbb{E}[\varepsilon_i^2] = \sigma^2 + \mathbf{z}_i^\top \alpha. \quad (2.174)$$

Sebab ε_i tidak dapat diamati secara langsung, akan digunakan residual OLS e_i sebagai *proxy*. Langkah selanjutnya adalah melakukan regresi *auxiliary* dengan menggunakan kuadrat residual sebagai variabel dependen, yaitu

$$e_i^2 = \gamma_0 + \mathbf{z}_i^\top \gamma + v_i, \quad (2.175)$$

dengan \mathbf{z}_i dapat berupa koordinat spasial, *lag* spasial dari variabel, atau fungsi jarak antar lokasi. Selanjutnya dikonstruksikan statistik uji. Didefinisikan vektor $\mathbf{f} = \mathbf{X}^T \mathbf{A} e^2$ dengan \mathbf{A} adalah matriks transformasi yang mengkodekan struktur spasial dan e^2 adalah vektor yang berisi kuadrat residual. Statistik Breusch-Pagan lokal kemudian didefinisikan sebagai

$$\text{BP}_{\text{Lokal}} = \frac{1}{2} \mathbf{f}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{f}. \quad (2.176)$$

Untuk menunjukkan bahwa statistik ini mengikuti distribusi χ^2 , perlu dibuktikan sifat-sifat distribusinya di bawah H_0 . Dengan asumsi regularitas dan normalitas galat, diperoleh tiga sifat penting, yaitu

- (i) nilai harapan dari \mathbf{f} adalah nol, yaitu $\mathbb{E}[\mathbf{f}] = \mathbf{0}$, karena di bawah H_0 terdapat $\mathbb{E}[e_i^2] = \sigma^2$ yang konstan untuk semua i ;
- (ii) matriks kovarians dari \mathbf{f} adalah $\text{Var}(\mathbf{f}) = 2\sigma^4(\mathbf{X}^\top \mathbf{X})$ ketika galat mengikuti distribusi normal; dan
- (iii) secara asimtotik, vektor \mathbf{f} konvergen dalam distribusi ke $\mathcal{N}(\mathbf{0}, 2\sigma^4(\mathbf{X}^\top \mathbf{X}))$.

Dengan menggunakan sifat-sifat di atas, dapat ditunjukkan konvergensi distribusi

statistik BP_{Lokal} , yaitu

$$BP_{\text{Lokal}} = \frac{1}{2} \mathbf{f}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{f}, \quad (2.177)$$

$$= \frac{1}{4\sigma^4} \mathbf{f}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{f} \cdot 2\sigma^4, \quad (2.178)$$

$$\xrightarrow{d} \frac{1}{2} \mathbf{Z}^\top \mathbf{Z}, \quad (2.179)$$

dengan $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$ adalah vektor acak normal standar berdimensi p .

Langkah terakhir adalah menggunakan sifat distribusi normal standar. Karena $\mathbf{Z}^\top \mathbf{Z} = \sum_{j=1}^p Z_j^2$ dan setiap Z_j^2 mengikuti distribusi χ^2 dengan satu derajat bebas, maka jumlahnya mengikuti $\chi^2(p)$. Oleh karena itu,

$$BP_{\text{Lokal}} \xrightarrow{d} \frac{1}{2} \chi^2(p). \quad (2.180)$$

Dengan normalisasi yang tepat, diperoleh hasil akhir bahwa $BP_{\text{Lokal}} \sim \chi^2(p)$ di bawah hipotesis nol H_0 . Dengan demikian, uji heterogenitas spasial dapat dilakukan dengan membandingkan nilai statistik BP_{Lokal} dengan nilai kritis dari distribusi χ^2 dengan derajat bebas p . ■

Selain uji heterogenitas spasial yang telah dibahas sebelumnya, terdapat beberapa uji hipotesis penting lainnya dalam konteks model GWR dan GTWR yang perlu dipertimbangkan untuk validasi model (Anselin, 1988; Fotheringham dkk., 2002).

Teorema 2.2.29 (Uji Signifikansi Koefisien Lokal GWR) *Untuk menguji signifikansi koefisien lokal $\beta_k(s_i)$ pada lokasi tertentu, dapat digunakan statistik uji*

$$t_k(s_i) = \frac{\hat{\beta}_k(s_i)}{\sqrt{\hat{\sigma}^2 \mathbf{e}_k^\top (\mathbf{X}^\top \mathbf{W}_i \mathbf{X})^{-1} \mathbf{e}_k}}, \quad (2.181)$$

dengan \mathbf{e}_k adalah vektor unit ke- k . Di bawah $H_0 : \beta_k(s_i) = 0$, statistik ini mengikuti distribusi t dengan derajat bebas $n_{\text{eff}} - p$, dengan n_{eff} adalah ukuran sampel efektif lokal.

Bukti. Mengikuti dari teorema interval konfidensi yang telah dibuktikan sebelumnya, estimator $\hat{\beta}_k(s_i)$ mengikuti distribusi normal asimtotik dengan mean $\beta_k(s_i)$ dan varians $\sigma^2 \mathbf{e}_k^T (\mathbf{X}^T \mathbf{W}_i \mathbf{X})^{-1} \mathbf{e}_k$. Dengan menstandarisasi dan menggunakan estimator konsisten $\hat{\sigma}^2$ untuk σ^2 , diperoleh statistik t yang dapat digunakan untuk pengujian hipotesis. Derajat bebas efektif $n_{\text{eff}} - p$ merefleksikan pengurangan informasi akibat pembobotan lokal (Fotheringham dkk., 2002). ■

Teorema 2.2.30 (Uji Stasioneritas Spasial) *Untuk menguji apakah koefisien regresi bersifat stasioner secara spasial (tidak bervariasi antar lokasi), dapat digunakan uji F dengan statistik*

$$F = \frac{(RSS_{OLS} - RSS_{GWR})/df_1}{RSS_{GWR}/df_2}, \quad (2.182)$$

dengan RSS_{OLS} dan RSS_{GWR} adalah jumlah kuadrat residual dari model OLS global dan GWR, sedangkan df_1 dan df_2 adalah derajat bebas yang sesuai. Di bawah H_0 bahwa koefisien stasioner, $F \sim F(df_1, df_2)$.

Uji ini didasarkan pada perbandingan kemampuan prediksi model global (OLS) dibandingkan model lokal (GWR). Jika tidak ada heterogenitas spasial yang signifikan, maka peningkatan kemampuan prediksi dari GWR tidak akan substansial. Statistik F mengukur apakah pengurangan RSS dari GWR secara statistik signifikan dibandingkan dengan kompleksitas tambahan yang diintroduksi oleh parameter lokal (Fotheringham dkk., 2002).

Teorema 2.2.31 (Diagnostik Model GWR) *Beberapa diagnostik penting untuk evaluasi model GWR meliputi:*

(a) **Akaike Information Criterion (AIC) terkoreksi:**

$$AIC_c = 2n \log(\hat{\sigma}) + n \log(2\pi) + n \left\{ \frac{n + \text{tr}(\mathbf{S})}{n - 2 - \text{tr}(\mathbf{S})} \right\}, \quad (2.183)$$

dengan \mathbf{S} adalah matriks hat dan $\text{tr}(\mathbf{S})$ adalah jejak matriks yang mengukur derajat bebas efektif.

(b) *Effective degrees of freedom:*

$$df = \text{tr}(\mathbf{S}), \quad (2.184)$$

yang mengukur kompleksitas model efektif.

(c) *Indeks determinasi lokal:*

$$R_{\text{lokal}}^2(s_i) = 1 - \frac{\text{RSS}_{\text{lokal}}(s_i)}{\text{TSS}_{\text{lokal}}(s_i)}, \quad (2.185)$$

yang mengukur kemampuan prediksi model di setiap lokasi.

Teorema 2.2.32 (Uji Multikolinearitas Lokal) Untuk mendeteksi multikolinearitas dalam konteks lokal, dapat digunakan indeks kondisi lokal

$$\kappa_i = \sqrt{\frac{\lambda_{\max}(\mathbf{X}^\top \mathbf{W}_i \mathbf{X})}{\lambda_{\min}(\mathbf{X}^\top \mathbf{W}_i \mathbf{X})}}, \quad (2.186)$$

dengan λ_{\max} dan λ_{\min} adalah nilai eigen terbesar dan terkecil dari matriks $\mathbf{X}^\top \mathbf{W}_i \mathbf{X}$. Nilai $\kappa_i > 30$ mengindikasikan masalah multikolinearitas yang serius di lokasi i .

Teorema 2.2.33 (Uji Autokorelasi Residual Spasial untuk GWR) Untuk menguji keberadaan autokorelasi spasial pada residual model GWR, dapat digunakan statistik Moran's I yang dimodifikasi:

$$I_{\text{GWR}} = \frac{n}{\sum_{i=1}^n \sum_{j=1}^n w_{ij}} \cdot \frac{\sum_{i=1}^n \sum_{j=1}^n w_{ij} e_i e_j}{\sum_{i=1}^n e_i^2}, \quad (2.187)$$

dengan e_i adalah residual GWR di lokasi i . Di bawah H_0 bahwa tidak ada autokorelasi spasial, I_{GWR} mengikuti distribusi normal asimtotik.

Pengujian-pengujian ini memberikan kerangka diagnostik yang komprehensif untuk memvalidasi model heterogenitas spasial dan memastikan bahwa asumsi-asumsi yang mendasari analisis GWR dan GTWR terpenuhi dengan baik.

2.3. Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) atau *artificial neural networks* (ANN) mempelajari pemetaan nonlinier dari ruang masukan ke keluaran melalui komposisi berlapis fungsi-fungsi sederhana (Bishop, 1995; Goodfellow, Bengio, & Courville, 2016; Haykin, 2009).

2.3.1. Model Dasar dan Notasi

Definisi 2.3.1 (Neuron Tiruan, (Haykin, 2009)) Diberikan $\mathbf{x} \in \mathbb{R}^p$, neuron menghasilkan

$$y = \phi(\mathbf{w}^\top \mathbf{x} + b), \quad (2.188)$$

dengan $\mathbf{w} \in \mathbb{R}^p$ adalah bobot, $b \in \mathbb{R}$ adalah bias, dan $\phi : \mathbb{R} \rightarrow \mathbb{R}$ adalah fungsi aktivasi nonlinier terdiferensiasi hampir di semua titik.

Definisi 2.3.2 (Jaringan Feedforward Multilapis (Bishop, 1995)) Untuk L lapisan tersembunyi, definisikan rekursi

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \quad \mathbf{a}^{(l)} = \phi^{(l)}(\mathbf{z}^{(l)}), \quad l = 1, \dots, L, \quad (2.189)$$

dengan $\mathbf{a}^{(0)} = \mathbf{x}$ dan keluaran akhir $\hat{y} = g(\mathbf{a}^{(L)})$. Parameter jaringan $\Theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^L$ dipelajari dari data.

Contoh 2.3.3 Misalkan $\mathbf{x} = (1, 2)^\top$, bobot $\mathbf{w} = (0.5, 0.25)^\top$, bias $b = 1$, dan $\varphi(z) = \tanh(z)$. Maka keluaran neuron adalah

$$y = \tanh(0.5 \cdot 1 + 0.25 \cdot 2 + 1) = \tanh(2) \approx 0.964.$$

2.3.2. Fungsi Aktivasi

Fungsi aktivasi memperkenalkan non-linearitas pada ANN. Tanpa fungsi aktivasi non-linear, jaringan saraf hanya merepresentasikan transformasi linear berlapis.

Definisi 2.3.4 (Fungsi Sigmoid) *Fungsi sigmoid didefinisikan sebagai*

$$\varphi(z) = \frac{1}{1 + e^{-z}}. \quad (2.190)$$

Contoh 2.3.5 Jika $z = -10$, maka

$$\varphi(-10) = \frac{1}{1 + e^{10}} \approx \frac{1}{1 + 22026.5} \approx 4.54 \times 10^{-5}.$$

Fungsi sigmoid sangat jenuh untuk $|z|$ besar: gradiennya mendekati nol.

Definisi 2.3.6 (Fungsi Tanh) *Fungsi tanh didefinisikan sebagai*

$$\varphi(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (2.191)$$

Contoh 2.3.7 Jika $z = 0$, maka

$$\varphi(0) = \tanh(0) = 0.$$

Jika $z = 2$, maka

$$\varphi(2) = \frac{e^2 - e^{-2}}{e^2 + e^{-2}} \approx \frac{7.389 - 0.135}{7.389 + 0.135} \approx \frac{7.254}{7.524} \approx 0.964.$$

Jika $z = -2$, maka

$$\varphi(-2) = \frac{e^{-2} - e^2}{e^{-2} + e^2} \approx \frac{0.135 - 7.389}{0.135 + 7.389} \approx \frac{-7.254}{7.524} \approx -0.964.$$

Fungsi tanh memiliki *range* $(-1, 1)$ dan gradien maksimum di sekitar $z = 0$.

Definisi 2.3.8 (Fungsi ReLU, (Nair & Hinton, 2010)) *Fungsi ReLU (Rectified Linear Unit) didefinisikan sebagai*

$$\varphi(z) = \max(0, z). \quad (2.192)$$

Contoh 2.3.9 Jika $z = -3$, maka $\varphi(z) = 0$. Jika $z = 2.5$, maka $\varphi(z) = 2.5$.

Definisi 2.3.10 (Fungsi Leaky ReLU) Fungsi *Leaky ReLU* mengatasi masalah dying ReLU dengan

$$\varphi(z) = \begin{cases} z, & z \geq 0, \\ \alpha z, & z < 0, \end{cases} \quad (2.193)$$

dengan $\alpha \in (0, 1)$ kecil, misalnya 0.01.

Contoh 2.3.11 Jika $z = -3$ dan $\alpha = 0.01$, maka $\varphi(z) = -0.03$. Jika $z = 2$, maka $\varphi(z) = 2$.

Definisi 2.3.12 (Fungsi GELU) Fungsi *Gaussian Error Linear Unit (GELU)* adalah

$$\varphi(z) = z \cdot \Phi(z), \quad (2.194)$$

dengan $\Phi(z)$ fungsi distribusi normal standar. Aproksimasi praktisnya adalah sebagai berikut.

$$\varphi(z) \approx 0.5z \left(1 + \tanh \left[\sqrt{\frac{2}{\pi}} (z + 0.0447z^3) \right] \right). \quad (2.195)$$

Contoh 2.3.13 Untuk $z = 1$, $\varphi(1) \approx 0.84$. Untuk $z = -1$, $\varphi(-1) \approx -0.16$.

Lema 2.3.14 (Sifat Fungsi Aktivasi) Fungsi aktivasi $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ yang digunakan pada jaringan saraf tiruan sebaiknya memenuhi:

- (i) φ non-linear,
- (ii) φ terdiferensiasi hampir di semua titik,
- (iii) φ tidak konstan, dan
- (iv) φ memiliki range memadai, tidak menyebabkan saturasi total.

Bukti.

- (i) *Nonlinearitas.* Jika φ linear, misalnya $\varphi(z) = az + b$, maka suatu multilayer perceptron dengan L lapisan menghasilkan

$$f(\mathbf{x}) = \mathbf{W}^{(L)} \varphi(\mathbf{W}^{(L-1)} \varphi(\dots \varphi(\mathbf{W}^{(1)} \mathbf{x}))) = \mathbf{W}' \mathbf{x} + \mathbf{b}',$$

dengan $\mathbf{W}' = \prod_{l=1}^L a \mathbf{W}^{(l)}$ dan \mathbf{b}' kombinasi linear dari \mathbf{b} . Komposisi tersebut tetap linear. Akibatnya, jaringan tidak lebih kuat daripada regresi linear biasa dan tidak mampu mengaproksimasi fungsi nonlinear, misalnya XOR.

- (ii) *Kediferensialan*. Algoritma *backpropagation* memerlukan turunan $\varphi'(z)$ untuk menghitung gradien fungsi kerugian L . Dengan notasi $\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$ dan $\mathbf{a}^{(l)} = \varphi(\mathbf{z}^{(l)})$, bentuk umum gradien adalah

$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \delta^{(l)} (\mathbf{a}^{(l-1)})^\top, \quad \delta^{(l)} = (\mathbf{W}^{(l+1)})^\top \delta^{(l+1)} \odot \varphi'(\mathbf{z}^{(l)}).$$

Jika $\varphi'(z)$ tidak terdefinisi pada banyak titik, maka gradien tidak dapat dihitung. Oleh karena itu, φ setidaknya harus terdiferensiasi hampir di semua titik. Fungsi seperti ReLU tetap sah karena ketakdiferensialan hanya terjadi pada $z = 0$ (himpunan titik ukuran nol).

- (iii) *Nonkonstansi*. Jika $\varphi(z) = c$ konstan, maka $\varphi'(z) = 0$ untuk semua z . Substitusi ke rumus backpropagation menghasilkan $\delta^{(l)} = \mathbf{0}$ untuk semua l , sehingga

$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \mathbf{0}.$$

Bobot tidak akan pernah diperbarui, sehingga proses pelatihan tidak berjalan.

- (iv) *Rentang keluaran yang memadai*. Jika φ cepat jenuh, misalnya sigmoid dengan limit $\varphi(z) \rightarrow 0$ untuk $z \rightarrow -\infty$ dan $\varphi(z) \rightarrow 1$ untuk $z \rightarrow +\infty$, maka untuk $|z|$ besar berlaku

$$\varphi'(z) = \varphi(z)(1 - \varphi(z)) \approx 0.$$

Akibatnya, besaran kesalahan terpropagasi $\delta^{(l)}$ mengecil secara eksponensial menuju nol (fenomena *vanishing gradient*). Hal ini dapat menggagalkan pembelajaran pada lapisan-lapisan dalam. Untuk mengatasinya, digunakan fungsi aktivasi dengan rentang yang lebih lebar atau gradien yang tidak menghilang, misalnya ReLU atau Leaky ReLU.



2.3.3. Fungsi Kerugian dan Kriteria Pembelajaran

Definisi 2.3.15 (Risiko Empiris, (Goodfellow dkk., 2016)) Diberikan data terawasi $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ dan model prediksi $\hat{y}_i = f_{\Theta}(\mathbf{x}_i)$, tujuan pembelajaran adalah meminimalkan

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n \ell(\hat{y}_i, y_i), \quad (2.196)$$

dengan fungsi kerugian ℓ dipilih sesuai dengan tugas yang dihadapi.

Contoh 2.3.16 Untuk regresi skalar, sebagai contoh digunakan $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$. Hal ini berarti jika $\hat{y} = 2.3$ dan $y = 1.5$, maka $\ell = \frac{1}{2}(0.8)^2 = 0.32$.

Contoh 2.3.17 Untuk klasifikasi biner, digunakan probabilitas prediksi $\hat{p} = \sigma(z)$ dan fungsi kerugian

$$\ell(\hat{p}, y) = -[y \log \hat{p} + (1 - y) \log(1 - \hat{p})], \quad y \in \{0, 1\}. \quad (2.197)$$

Fungsi ini merupakan entropi silang antara distribusi target $p(y)$ dan prediksi \hat{p} , yaitu

$$\begin{aligned} H(p, \hat{p}) &= - \sum_{y \in \{0, 1\}} p(y) \log \hat{p}(y) \\ &= -[p(1) \log \hat{p} + p(0) \log(1 - \hat{p})]. \end{aligned} \quad (2.198)$$

Dengan $p(1) = y$ dan $p(0) = 1 - y$, diperoleh bentuk di atas.

Sebagai contoh, jika $y = 1$ dan $\hat{p} = 0.9$, maka $\ell = -\log 0.9 \approx 0.105$ (kerugian kecil). Jika $y = 0$ dan $\hat{p} = 0.9$, maka $\ell = -\log 0.1 \approx 2.303$ (kerugian besar).

Contoh 2.3.18 Untuk klasifikasi multikelas, dengan logit $\mathbf{z} \in \mathbb{R}^K$, softmax $\hat{p}_k =$

$\exp(z_k) / \sum_{j=1}^K \exp(z_j)$ dan

$$\ell(\hat{\mathbf{p}}, y) = -\log \hat{p}_y, \quad y \in \{1, \dots, K\}.$$

Fungsi ini juga merupakan entropi silang antara distribusi target $p(y)$ dan prediksi $\hat{\mathbf{p}}$:

$$\begin{aligned} H(p, \hat{\mathbf{p}}) &= -\sum_{k=1}^K p(k) \log \hat{p}_k \\ &= -\log \hat{p}_y, \end{aligned} \quad (2.199)$$

dengan $p(k) = 1$ jika $k = y$ dan 0 selainnya.

Sebagai contoh, untuk $K = 3$ kelas, jika $\mathbf{z} = (2.0, 1.0, 0.1)$, maka softmax $\hat{\mathbf{p}} \approx (0.659, 0.242, 0.099)$. Jika label benar $y = 1$, maka $\ell = -\log 0.659 \approx 0.417$. Jika $y = 3$, maka $\ell = -\log 0.099 \approx 2.313$.

2.3.4. Pembelajaran dengan Propagasi Mundur

Teorema 2.3.19 (Aturan Rantai pada Jaringan) Misalkan keluaran akhir didefinisikan sebagai komposisi $y = f^{(L)} \circ f^{(L-1)} \circ \dots \circ f^{(1)}(\mathbf{x})$. Maka turunan terhadap parameter pada lapisan ke- l memenuhi

$$\frac{\partial y}{\partial \mathbf{W}^{(l)}} = \frac{\partial f^{(L)}}{\partial f^{(L-1)}} \cdots \frac{\partial f^{(l+1)}}{\partial f^{(l)}} \frac{\partial f^{(l)}}{\partial \mathbf{W}^{(l)}}. \quad (2.200)$$

Interpretasi teorema ini adalah bahwa gradien dihitung dengan mengalikan turunan setiap lapisan dari keluaran ke lapisan yang diinginkan. Hal ini berarti akan dihitung efek perubahan bobot pada lapisan tersebut terhadap keluaran akhir melalui semua lapisan berikutnya. Apabila efek ini kecil, maka gradien akan mengecil secara eksponensial (fenomena *vanishing gradient*). Apabila efek ini besar, maka gradien akan membesar secara eksponensial (fenomena *exploding gradient*). Secara umum, pembelajaran menginginkan untuk memperbaiki bobot yang memiliki efek signifikan terhadap keluaran, tetapi tidak terlalu besar sehingga menyebabkan

ketidakstabilan.

Teorema 2.3.20 (Propagasi Mundur atau *Backpropagation*, (Goodfellow dkk., 2016)) Dengan notasi $\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$, $\mathbf{a}^{(l)} = \varphi(\mathbf{z}^{(l)})$, dan $\ell(\hat{y}, y)$ fungsi kerugian, gradien terhadap bobot $\mathbf{W}^{(l)}$ diberikan oleh

$$\frac{\partial \ell}{\partial \mathbf{W}^{(l)}} = \delta^{(l)} (\mathbf{a}^{(l-1)})^\top, \quad (2.201)$$

dengan $\delta^{(l)} = (\mathbf{W}^{(l+1)})^\top \delta^{(l+1)} \odot \varphi'(\mathbf{z}^{(l)})$. Operator \odot adalah perkalian elemen-per-elemen atau Hadamard product.

Contoh 2.3.21 Misalkan jaringan 2 lapis dengan input $\mathbf{x} = (x_1, x_2)^\top$, bobot

$$\mathbf{W}^{(1)} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix},$$

bias $\mathbf{b}^{(1)} = (0, 0)^\top$, fungsi aktivasi $\varphi(z) = \tanh(z)$, dan lapisan output linear $y = \mathbf{w}^{(2)\top} \mathbf{a}^{(1)}$ dengan $\mathbf{w}^{(2)} = (1, 1)^\top$. Target $y^* = 1$, kerugian $\ell(y, y^*) = \frac{1}{2}(y - y^*)^2$.

(a) *Forward*:

$$\begin{aligned} \mathbf{z}^{(1)} &= \mathbf{W}^{(1)} \mathbf{x} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \mathbf{a}^{(1)} &= \tanh(\mathbf{z}^{(1)}) \\ y &= (1, 1) \cdot \mathbf{a}^{(1)} \end{aligned}$$

(b) *Backward*:

$$\begin{aligned} \delta^{(2)} &= \frac{\partial \ell}{\partial y} = y - y^* \\ \delta^{(1)} &= \mathbf{w}^{(2)} \cdot \delta^{(2)} \odot \varphi'(\mathbf{z}^{(1)}) \\ \frac{\partial \ell}{\partial \mathbf{W}^{(1)}} &= \delta^{(1)} (\mathbf{x})^\top \end{aligned}$$

Dengan substitusi nilai, gradien dapat dihitung secara eksplisit untuk setiap para-

meter.

2.3.5. Optimisasi Parameter

Dalam pelatihan jaringan saraf, parameter Θ dioptimalkan untuk meminimalkan fungsi risiko $L(\Theta)$. Beberapa algoritma optimisasi yang akan dibahas adalah *Gradient Descent* (GD), *Stochastic Gradient Descent* (SGD), dan *Adaptive Moment Estimation* (Adam).

Definisi 2.3.22 (*Gradient Descent*) Diberikan fungsi risiko empiris

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n \ell(f_{\Theta}(\mathbf{x}_i), y_i),$$

pembaruan parameter dengan laju belajar $\eta > 0$ adalah

$$\Theta_{t+1} = \Theta_t - \eta \nabla_{\Theta} L(\Theta_t). \quad (2.202)$$

Metode ini konvergen ke minimum lokal untuk fungsi non-konveks seperti jaringan saraf.

Intuisi dari metode berbasis gradien ini adalah memperbarui parameter model ke arah yang menurunkan fungsi kerugian paling cepat. Pada setiap iterasi, gradien $\nabla_{\Theta} L(\Theta)$ menunjukkan arah perubahan parameter yang paling efektif untuk mengurangi kerugian. Dengan memilih laju pembelajaran η yang sesuai, proses optimisasi bergerak menuju minimum lokal dari fungsi kerugian. Jika $L(\Theta)$ konveks, metode ini menjamin konvergensi ke minimum global. Namun, pada kasus non-konveks seperti jaringan saraf, *gradient descent* tetap efektif menemukan solusi yang baik secara empiris. Variasi seperti SGD dan Adam mempercepat konvergensi dan meningkatkan generalisasi dengan memanfaatkan batch acak dan penyesuaian adaptif terhadap gradien.

Contoh 2.3.23 Misalkan fungsi kerugian $L(w) = \frac{1}{2}(w - 3)^2$. Gradiennya adalah $\nabla_w L(w) = w - 3$. Dengan laju belajar $\eta = 0.1$ dan inisialisasi $w_0 = 0$, maka iterasi

gradient descent adalah sebagai berikut.

$$\begin{aligned} w_1 &= w_0 - 0.1 \cdot (w_0 - 3) = 0 - 0.1 \cdot (-3) = 0.3 \\ w_2 &= w_1 - 0.1 \cdot (w_1 - 3) = 0.3 - 0.1 \cdot (-2.7) = 0.57 \\ w_3 &= 0.57 - 0.1 \cdot (0.57 - 3) = 0.813 \\ &\vdots \end{aligned}$$

Nilai w_t akan konvergen menuju 3.

Definisi 2.3.24 (Stochastic Gradient Descent (SGD)) *SGD adalah metode optimisasi yang memperbarui parameter model menggunakan rata-rata gradien dari batch kecil acak berukuran m pada setiap iterasi. Batch adalah sekumpulan sampel yang digunakan untuk menghitung gradien, serta batch kecil ini diambil secara acak dari dataset. Pembaruan parameter dilakukan sebagai berikut.*

$$\Theta_{t+1} = \Theta_t - \eta \frac{1}{|\mathcal{B}_t|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{B}_t} \nabla_{\Theta} \ell(f_{\Theta}(\mathbf{x}_i), y_i), \quad (2.203)$$

dengan \mathcal{B}_t adalah batch kecil pada iterasi ke- t , η adalah laju pembelajaran, dan ℓ adalah fungsi kerugian. Metode ini sangat efisien untuk data berukuran besar dan memberikan regularisasi implisit melalui pengacakan batch.

Contoh 2.3.25 Misalkan fungsi kerugian $L(w) = \frac{1}{2}(w - 3)^2$ dan data terdiri dari $n = 1000$ sampel, masing-masing x_i dengan target $y_i = 3$ untuk semua i . Dengan batch kecil berukuran $m = 10$, pada setiap iterasi SGD dilakukan langkah berikut.

- (a) Pilih acak $m = 10$ sampel dari data, misal indeks $\mathcal{B}_t = \{i_1, \dots, i_{10}\}$.
- (b) Hitung rata-rata gradien pada batch:

$$\nabla_w L_{\mathcal{B}_t}(w) = \frac{1}{m} \sum_{j=1}^m (w - y_{i_j}) = w - 3$$

(karena semua $y_{i_j} = 3$).

(c) Perbarui parameter dengan laju belajar η :

$$w_{t+1} = w_t - \eta(w_t - 3)$$

(d) Ulangi langkah (a)–(c) hingga w_t konvergen ke 3.

Dengan batch acak, setiap iterasi hanya menggunakan sebagian kecil data, sehingga proses lebih efisien dan tetap menuju minimum global $w = 3$.

Definisi 2.3.26 (Adaptive Moment Estimation (Adam)) Adam adalah algoritma optimisasi berbasis gradien yang menggabungkan momentum (rata-rata bergerak gradien) dan skala adaptif (rata-rata kuadrat gradien). Adam memperbarui parameter dengan estimasi momen pertama dan kedua yang dikoreksi bias. Dengan gradien $\mathbf{g}_t = \nabla_{\Theta} L_t(\Theta_t)$, inisialisasi $\mathbf{m}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$, lalu untuk setiap iterasi $t \geq 1$:

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t, \quad (2.204)$$

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) (\mathbf{g}_t \odot \mathbf{g}_t), \quad (2.205)$$

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t}, \quad \hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t}, \quad (2.206)$$

$$\Theta_{t+1} = \Theta_t - \eta \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \varepsilon}. \quad (2.207)$$

Parameter umum: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$.

Contoh 2.3.27 Misalkan gradien pada iterasi pertama $\mathbf{g}_1 = (0.1, -0.2)$, inisialisasi $\mathbf{m}_0 = (0, 0)$, $\mathbf{v}_0 = (0, 0)$, $\eta = 0.01$. Maka:

$$\mathbf{m}_1 = 0.9 \cdot (0, 0) + 0.1 \cdot (0.1, -0.2) = (0.01, -0.02),$$

$$\mathbf{v}_1 = 0.999 \cdot (0, 0) + 0.001 \cdot (0.01, 0.04) = (0.00001, 0.00004),$$

$$\hat{\mathbf{m}}_1 = (0.01, -0.02) / (1 - 0.9^1) = (0.1, -0.2),$$

$$\hat{\mathbf{v}}_1 = (0.00001, 0.00004) / (1 - 0.999^1) = (0.01, 0.04),$$

$$\Theta_2 = \Theta_1 - 0.01 \cdot \frac{(0.1, -0.2)}{(\sqrt{0.01}, \sqrt{0.04}) + 10^{-8}} = \Theta_1 - 0.01 \cdot (1, -1).$$

Jadi, parameter diperbarui dengan langkah adaptif pada setiap komponen.

2.3.6. Jaringan Saraf Tiruan sebagai Aproksimasi Universal

Salah satu hasil teoretis paling penting tentang jaringan saraf tiruan adalah Teorema Aproksimasi Universal (UAT). Teorema ini menyatakan bahwa jaringan saraf dengan satu lapisan tersembunyi yang cukup besar dapat mengaproksimasi fungsi kontinu pada himpunan kompak dengan presisi setinggi apapun, asalkan fungsi aktivasi memenuhi syarat tertentu.

Teorema 2.3.28 (Teorema Aproksimasi Universal (UAT), (Cybenko, 1989)) *Jika $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ adalah fungsi sigmoid kontinu tak konstan (misal, $\varphi(z) = 1/(1 + e^{-z})$ atau $\varphi(z) = \tanh(z)$), maka untuk setiap fungsi kontinu f pada $[0, 1]^p$ dan setiap $\varepsilon > 0$, ada jaringan satu lapis dengan jumlah neuron cukup banyak sehingga*

$$\sup_{\mathbf{x} \in [0,1]^p} |f(\mathbf{x}) - g(\mathbf{x})| < \varepsilon,$$

dengan $g(\mathbf{x}) = \sum_{j=1}^m c_j \varphi(\mathbf{w}_j^\top \mathbf{x} + b_j)$.

2.4. Jaringan Saraf Graf

Jaringan Saraf Graf (*Graph Neural Networks*, GNN) merupakan kelas arsitektur pembelajaran representasi yang dirancang khusus untuk data terstruktur dalam bentuk graf. Berbeda dengan jaringan saraf tiruan (JST) yang bekerja pada vektor atau matriks, GNN memanfaatkan informasi topologi melalui graf $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ dengan simpul \mathcal{V} dan sisi \mathcal{E} .

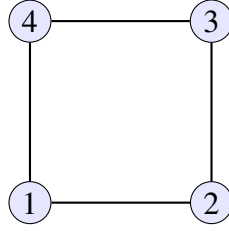
2.4.1. Dasar Graf dan Laplacian

Definisi 2.4.1 (Graf dan Matriks Ketetanggaan) *Graf $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ dengan $|\mathcal{V}| = n$ direpresentasikan oleh:*

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{jika } (i, j) \in \mathcal{E}, \\ 0 & \text{lainnya,} \end{cases} \quad (2.208)$$

disebut matriks ketetanggaan. Matriks derajat didefinisikan sebagai $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$, dengan $d_i = \sum_j A_{ij}$.

Contoh 2.4.2 Graf $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ dengan $|\mathcal{V}| = 4$ dan sisi $\mathcal{E} = \{(1, 2), (2, 3), (3, 4), (4, 1)\}$ dapat divisualisasikan sebagai berikut.



Matriks ketetanggaan \mathbf{A} untuk graf di atas adalah

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Dengan kata lain, matriks derajatnya $\mathbf{D} = \text{diag}(2, 2, 2, 2)$.

Definisi 2.4.3 (Graf Laplacian, (Chung, 1997)) Graf Laplacian adalah $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Normalisasi simetris diberikan oleh

$$\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}. \quad (2.209)$$

Graf Laplacian merepresentasikan struktur graf dan digunakan dalam berbagai algoritma GNN.

Contoh 2.4.4 Misalkan graf \mathcal{G} dengan 4 simpul dan sisi $\mathcal{E} = \{(1, 2), (2, 3), (3, 4), (4, 1)\}$ (graf siklus). Matriks ketetanggaannya adalah

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

dan matriks derajatnya adalah

$$\mathbf{D} = \text{diag}(2, 2, 2, 2).$$

Graf Laplacian-nya dapat dihitung sebagai

$$\mathbf{L} = \mathbf{D} - \mathbf{A} = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}.$$

Sehingga, dengan normalisasi simetris didapatkan

$$\mathbf{L}_{\text{sym}} = \mathbf{I} - \frac{1}{2}\mathbf{A},$$

karena $\mathbf{D}^{-1/2} = \text{diag}(1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2})$.

Teorema 2.4.5 (Teorema Spektral Laplacian, (Chung, 1997)) *Matriks Laplacian \mathbf{L} adalah simetris dan positif semidefinit, sehingga dapat didekomposisi dengan dekomposisi nilai eigen sebagai berikut.*

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{2.210}$$

dengan \mathbf{U} ortogonal dan $\mathbf{\Lambda}$ diagonal berisi nilai eigen $\lambda_1, \dots, \lambda_n \geq 0$. Selalu berlaku $\lambda_1 = 0$ dengan vektor eigen $\mathbf{1}$.

2.4.2. Kerangka Penyampaian Pesan pada Jaringan Saraf

Definisi 2.4.6 (Message Passing Neural Networks (MPNN), (Bronstein, Bruna, Cohen, & Veličković, 2021)) *Misalkan graf $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ dengan \mathcal{V} adalah himpunan simpul dan \mathcal{E} adalah himpunan sisi. Misalkan pula \mathcal{N}_u adalah ketetanggaan simpul $u \in \mathcal{V}$. Lebih lanjut, misalkan \mathbf{x}_u adalah fitur simpul u dan \mathbf{e}_{uv} adalah fitur*

sisi $(u, v) \in \mathcal{E}$. Suatu lapisan MPNN didefinisikan sebagai berikut.

$$\mathbf{h}_u = \varphi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{uv}), \right) \quad (2.211)$$

dengan φ dan ψ adalah fungsi yang terdiferensiasi (seperti jaringan saraf) dan \bigoplus adalah operator agregasi yang bersifat komutatif dan asosiatif (seperti penjumlahan atau rata-rata) atau permutation invariant.

Kerangka MPNN ini sangat fleksibel dan dapat disesuaikan dengan berbagai jenis data graf. Fungsi ψ dapat dirancang untuk menangkap interaksi spesifik antara simpul dan sisi, sedangkan fungsi φ dapat berupa jaringan saraf yang kompleks untuk pembaruan fitur. Operator agregasi \bigoplus memastikan bahwa model tetap invariant terhadap permutasi tetangga, yang penting dalam konteks graf. Beberapa arsitektur umum GNN, seperti *Graph Convolutional Networks* (GCN) dan *Graph Attention Networks* (GAT), dapat dianggap sebagai kasus khusus dari kerangka MPNN ini.

2.4.3. Arsitektur Umum dalam GNN

Definisi 2.4.7 (Graph Convolutional Networks (GCN), (Kipf & Welling, 2017))

GCN mendefinisikan operasi propagasi lapisan ke- l sebagai

$$\mathbf{H}^{(l+1)} = \sigma \left(\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(l)} \mathbf{W}^{(l)} \right), \quad (2.212)$$

dengan:

- $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ matriks ketetanggaan dengan self-loop,
- $\tilde{\mathbf{D}} = \text{diag}(\sum_j \tilde{A}_{ij})$ matriks derajat dari $\tilde{\mathbf{A}}$,
- $\mathbf{H}^{(l)} \in \mathbb{R}^{n \times d_l}$ representasi simpul pada lapisan l ,
- $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ bobot terlatih, dan
- $\sigma(\cdot)$ fungsi aktivasi non-linear.

GCN merupakan kasus khusus dari kerangka MPNN dengan:

- Fitur sisi diabaikan (e_{uv} tidak digunakan).
- Fungsi pesan $\psi(\mathbf{x}_u, \mathbf{x}_v, e_{uv}) = \mathbf{x}_v$ (hanya mengambil fitur tetangga).
- Operator agregasi \oplus berupa penjumlahan atau rata-rata tertimbang (melalui normalisasi Laplacian).
- Fungsi update φ adalah komposisi linear dan aktivasi non-linear: $\varphi(\mathbf{x}_u, \text{AGG}) = \sigma(\text{AGG} \cdot \mathbf{W})$.

Secara eksplisit, propagasi GCN dapat ditulis sebagai

$$\mathbf{h}_u^{(l+1)} = \sigma \left(\sum_{v \in \mathcal{N}_u \cup \{u\}} \frac{1}{\sqrt{d_u d_v}} \mathbf{h}_v^{(l)} \mathbf{W}^{(l)} \right),$$

dengan d_u derajat simpul u dan σ fungsi aktivasi. Matriks normalisasi simetris $\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2}$ merepresentasikan agregasi pesan dari tetangga dengan bobot yang sesuai, sehingga GCN adalah MPNN dengan agregasi rata-rata dan update linear.

Definisi 2.4.8 (Graph Attention Networks (GAT), (Veličković dkk., 2018)) GAT mengganti normalisasi derajat dengan mekanisme perhatian. Koefisien perhatian untuk sisi (i, j) adalah

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_k]))}, \quad (2.213)$$

dengan:

- $\mathbf{h}_i \in \mathbb{R}^d$ vektor fitur simpul i ,
- \mathbf{W} bobot transformasi linear,
- $\mathbf{a} \in \mathbb{R}^{2d'}$ vektor bobot perhatian, dan
- \parallel operator konkatenasi.

Pembaruan fitur simpul adalah

$$\mathbf{h}'_i = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W} \mathbf{h}_j \right). \quad (2.214)$$

GAT merupakan kasus khusus dari kerangka MPNN dengan spesifikasi berikut.

- Fitur sisi e_{uv} tidak digunakan (atau dapat diabaikan).
- Fungsi pesan $\psi(\mathbf{x}_u, \mathbf{x}_v, e_{uv}) = \mathbf{W} \mathbf{x}_v$ (transformasi linear fitur tetangga).
- Operator agregasi \oplus berupa penjumlahan tertimbang, dengan bobot agregasi α_{uv} diperoleh dari mekanisme perhatian (attention) yang bersifat permutation invariant.
- Fungsi update φ adalah komposisi linear dan aktivasi non-linear: $\varphi(\mathbf{x}_u, \text{AGG}) = \sigma(\text{AGG})$.

Secara formal, pembaruan fitur simpul u pada GAT dapat ditulis dalam notasi MPNN sebagai

$$\mathbf{h}'_u = \sigma \left(\sum_{v \in \mathcal{N}_u} \alpha_{uv} \psi(\mathbf{x}_u, \mathbf{x}_v, e_{uv}) \right) = \sigma \left(\sum_{v \in \mathcal{N}_u} \alpha_{uv} \mathbf{W} \mathbf{x}_v \right), \quad (2.215)$$

dengan α_{uv} adalah skor perhatian yang bergantung pada fitur \mathbf{x}_u dan \mathbf{x}_v melalui fungsi $\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W} \mathbf{x}_u \parallel \mathbf{W} \mathbf{x}_v])$ dan normalisasi softmax. Dengan demikian, GAT adalah MPNN dengan agregasi penjumlahan tertimbang dan fungsi pesan linear dengan bobot agregasi ditentukan secara adaptif oleh mekanisme perhatian.

Definisi 2.4.9 (GraphSAGE, (Hamilton, Ying, & Leskovec, 2017)) GraphSAGE atau Graph Sample and Aggregation mendefinisikan pembaruan simpul i sebagai

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\mathbf{W}^{(l)} \cdot \text{AGGREGATE} \left(\{\mathbf{h}_i^{(l)}\} \cup \{\mathbf{h}_j^{(l)} : j \in \mathcal{N}(i)\} \right) \right), \quad (2.216)$$

dengan AGGREGATE fungsi agregasi non-linear yang bisa berupa mean, max-pooling, atau LSTM.

2.4.4. Kekuatan Representasi GNN

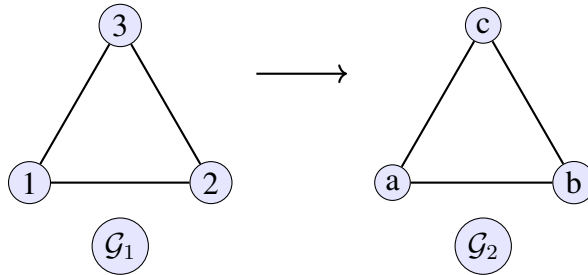
Definisi 2.4.10 (Isomorfisme Graf) Dua graf $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ dan $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ disebut isomorfik jika terdapat bijeksi $\pi : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ sehingga

$$(i, j) \in \mathcal{E}_1 \iff (\pi(i), \pi(j)) \in \mathcal{E}_2. \quad (2.217)$$

Sebuah fungsi f dikatakan invarian terhadap isomorfisme graf jika $f(\mathcal{G}_1) = f(\mathcal{G}_2)$ untuk setiap graf isomorfik $\mathcal{G}_1 \cong \mathcal{G}_2$.

Contoh 2.4.11 Misalkan \mathcal{G}_1 adalah graf dengan simpul $\{1, 2, 3\}$ dan sisi $\{(1, 2), (2, 3), (3, 1)\}$ (graf segitiga). Ambil \mathcal{G}_2 dengan simpul $\{a, b, c\}$ dan sisi $\{(a, b), (b, c), (c, a)\}$. Bijeksi π dapat dipilih misal $\pi(1) = a$, $\pi(2) = b$, $\pi(3) = c$. Maka \mathcal{G}_1 dan \mathcal{G}_2 isomorfik.

Sebagai contoh fungsi invarian, jumlah simpul $|\mathcal{V}|$ dan jumlah sisi $|\mathcal{E}|$ adalah invarian terhadap isomorfisme graf. Demikian juga, spektrum Laplacian graf \mathbf{L} adalah invarian terhadap isomorfisme.



Gambar 2.6 Ilustrasi dua graf isomorfik, yaitu \mathcal{G}_1 dan \mathcal{G}_2 memiliki struktur yang sama walaupun label simpul berbeda. (Sumber: Dokumen penulis)

Definisi 2.4.12 (Tes Weisfeiler–Lehman (WL) Satu Dimensi) Algoritma WL satu dimensi, juga dikenal sebagai color refinement, merupakan prosedur iteratif untuk membedakan graf berdasarkan struktur lokal simpul. Pada setiap iterasi, label simpul $h_i^{(t)}$ diperbarui dengan cara:

$$h_i^{(t+1)} = \text{HASH}\left(h_i^{(t)}, \{h_j^{(t)} : j \in \mathcal{N}(i)\}\right), \quad (2.218)$$

dimulai dari label awal $h_i^{(0)}$. Dua graf dikatakan dapat dibedakan oleh tes WL

jika multiset label akhirnya berbeda. Dalam konteks pembelajaran mesin, tes WL memberikan kerangka formal untuk mengukur kemampuan model, seperti GNN, dalam membedakan struktur graf yang berbeda melalui proses agregasi informasi dari tetangga.

Fungsi HASH pada algoritma WL berperan sebagai pemetaan yang menggabungkan label simpul saat ini dengan label-label tetangganya menjadi label baru yang unik. Intuisi utamanya adalah memastikan bahwa jika dua simpul memiliki lingkungan yang berbeda, hasil HASH juga berbeda. Dalam implementasi praktis, HASH dapat berupa fungsi injektif seperti pengkodean *string*, *concatenation*, atau fungsi neural yang mampu membedakan multiset masukan secara efektif.

Teorema 2.4.13 (Keterbatasan GNN Berbasis Agregasi, (Xu, Hu, Leskovec, & Jegelka, 2019)) *GNN berbasis agregasi dengan pembaruan umum*

$$h_i^{(l+1)} = \varphi \left(W^{(l)} \cdot \text{AGGREGATE}(\{h_i^{(l)}\} \cup \{h_j^{(l)} : j \in \mathcal{N}(i)\}) \right) \quad (2.219)$$

tidak lebih kuat daripada tes WL satu dimensi dalam membedakan graf yang tidak isomorfik. Artinya, kemampuan GNN untuk membedakan struktur graf dibatasi oleh kekuatan agregasi lokal yang serupa dengan tes WL.

Teorema 2.4.14 (Hubungan dengan Tes WL, (Xu dkk., 2019)) *Jika fungsi agregasi AGGREGATE bersifat injektif terhadap multiset, maka GNN memiliki kekuatan representasi setara dengan tes WL satu dimensi. Dengan kata lain, GNN dapat membedakan graf sejauh tes WL mampu membedakannya, asalkan agregasi dan pembaruan fitur dilakukan secara unik untuk setiap lingkungan simpul.*

2.5. Pembelajaran Representasi

Pembelajaran representasi (*representation learning*) bertujuan mempelajari transformasi dari data mentah ke bentuk fitur yang lebih sesuai untuk tugas pembelajaran. Secara formal, algoritma pembelajaran representasi mencari transformasi ϕ yang memetakan data observasi x ke representasi $h = \phi(x)$, sehingga prediktor sederhana f dapat memodelkan target y dari h secara efisien.

Pembelajaran representasi merupakan cabang pembelajaran mesin yang berfokus pada pembentukan representasi data yang memudahkan ekstraksi informasi penting untuk klasifikasi atau prediksi. Pendekatan ini menggantikan rekayasa fitur manual dengan proses otomatis yang mempelajari struktur intrinsik data. Dengan representasi yang baik, model dapat menangkap faktor penyebab utama data dan mengabaikan variasi yang tidak relevan terhadap tugas.

Tujuan utama pembelajaran representasi adalah memisahkan faktor-faktor variasi yang mendasari data dan merepresentasikannya agar lebih mudah dipisahkan untuk tugas prediksi berikutnya. Representasi yang baik tidak hanya memampatkan informasi, tetapi juga mengorganisasi struktur data sehingga faktor penting dapat diakses secara eksplisit oleh model.

Dalam pembelajaran mendalam, pembelajaran representasi memegang peran sentral karena memungkinkan sistem secara otomatis menemukan representasi yang diperlukan untuk deteksi fitur atau klasifikasi langsung dari data mentah. Proses ini menghasilkan vektor representasi $\phi(x)$ yang memuat informasi relevan dengan tugas, sekaligus menekan dimensi dan variasi yang tidak diperlukan. Representasi yang baik memudahkan fungsi f sederhana (misalnya regresi linear atau *softmax*) untuk mencapai kinerja tinggi tanpa rekayasa fitur tambahan.

Secara ringkas, pembelajaran representasi merupakan kerangka umum untuk

- (a) mempelajari pemetaan ϕ dari data mentah x ke ruang fitur laten h ,
- (b) menstrukturkan ulang informasi agar faktor variasi yang relevan menjadi terpisah, dan
- (c) memfasilitasi pembelajaran fungsi prediksi sederhana f di ruang laten.

Pendekatan ini menjadi dasar konseptual bagi jaringan saraf tiruan modern dan seluruh keluarga model pembelajaran mendalam.

Definisi 2.5.1 (Sifat Representasi yang Baik) Menurut Goodfellow dkk. (2016), representasi yang baik adalah transformasi $\phi(x)$ dari data mentah x ke ruang laten yang memenuhi sifat-sifat berikut.

- (a) *Invariansi, yaitu Representasi tidak berubah terhadap variasi tak relevan pada input, seperti translasi, rotasi, atau gangguan acak yang tidak memengaruhi makna semantik.*
- (b) *Disentanglement, yaitu Setiap dimensi dalam ruang laten memuat faktor penyebab utama yang terpisah, sehingga informasi penting tidak tercampur.*
- (c) *Efisiensi, yaitu Representasi bersifat jarang (sparse) atau berdimensi rendah, sehingga memudahkan penyimpanan, pemrosesan, dan generalisasi.*
- (d) *Transferabilitas, yaitu Representasi dapat digunakan ulang untuk berbagai tugas berbeda tanpa pelatihan ulang yang ekstensif.*

Suatu transformasi ϕ dikatakan menghasilkan representasi yang baik jika:

- (a) memisahkan faktor-faktor variasi utama dalam data sehingga fungsi target f dapat dipelajari secara sederhana di ruang laten,
- (b) membuat prediksi menjadi lebih mudah dan stabil terhadap perubahan kecil pada input yang tidak relevan, dan
- (c) memungkinkan generalisasi dan transfer ke tugas baru dengan sedikit data berlabel.

2.5.1. Landasan Teoretis Representasi

Definisi 2.5.2 (Invariansi dan Equivariansi, Goodfellow dkk., 2016) *Suatu representasi $\phi : \mathcal{X} \rightarrow \mathcal{Z}$ dikatakan invarian terhadap grup transformasi G jika memenuhi*

$$\phi(g \cdot x) = \phi(x), \quad \forall g \in G, x \in \mathcal{X}. \quad (2.220)$$

Sebaliknya, ϕ disebut equivariant terhadap G jika terdapat aksi linear T_g pada \mathcal{Z} sehingga

$$\phi(g \cdot x) = T_g \phi(x). \quad (2.221)$$

Representasi yang baik seharusnya invarian terhadap faktor-faktor yang tidak memengaruhi konsep target, seperti translasi kecil pada citra atau perubahan

kondisi pencahayaan. Invariansi menjadi kunci agar model tidak sensitif terhadap variasi yang tidak bermakna secara semantik.

Contoh 2.5.3 Pada klasifikasi digit MNIST, translasi kecil pada posisi angka tidak boleh mengubah representasi laten digit tersebut. Dalam jaringan konvolusional, fungsi ϕ dirancang agar bersifat *equivariant* terhadap translasi, sehingga pergeseran lokasi pada ruang input hanya memindahkan posisi aktivasi tanpa mengubah nilainya.

Definisi 2.5.4 (Hipotesis Manifold, Bengio, Courville, & Vincent, 2013) *Misalkan data $x \in \mathbb{R}^D$ terdistribusi di sekitar Manifold berdimensi rendah $\mathcal{M} \subset \mathbb{R}^D$. Hipotesis Manifold menyatakan bahwa terdapat pemetaan kontinu*

$$\phi : \mathbb{R}^D \rightarrow \mathbb{R}^d, \quad d \ll D, \quad (2.222)$$

yang memparametrisasi koordinat intrinsik Manifold \mathcal{M} , sehingga setiap titik x dapat direpresentasikan oleh koordinat laten $\phi(x)$.

Manifold adalah konsep matematis yang merepresentasikan himpunan titik-titik dalam ruang berdimensi tinggi yang secara lokal menyerupai ruang Euclidean berdimensi lebih rendah. Dalam konteks pembelajaran mesin, Manifold digunakan untuk menjelaskan bahwa data berdimensi tinggi sebenarnya terletak pada atau di sekitar permukaan berdimensi rendah di dalam ruang fitur aslinya. Misalnya, meskipun gambar digital memiliki ribuan piksel (dimensi tinggi), variasi utama seperti pose, pencahayaan, atau ekspresi wajah hanya membentuk beberapa derajat kebebasan (dimensi rendah).

Hipotesis Manifold menyatakan bahwa data nyata tidak tersebar secara acak di seluruh ruang berdimensi tinggi, melainkan terkonsentrasi di sekitar Manifold berdimensi rendah yang dibentuk oleh faktor-faktor variasi utama. Oleh karena itu, pembelajaran representasi yang efektif bertujuan menemukan koordinat intrinsik Manifold ini, sehingga model dapat memisahkan faktor penyebab utama dan mengabaikan variasi yang tidak relevan. Pendekatan ini mendasari banyak metode

modern, seperti *autoencoder*, pembelajaran berbasis graf, dan pembelajaran semi-terawasi, yang semuanya memanfaatkan struktur Manifold untuk meningkatkan generalisasi dan efisiensi model.

Teorema 2.5.5 (Hipotesis Manifold dan Invariansi, Goodfellow dkk., 2016) *Jika data $x \in \mathbb{R}^D$ berada pada Manifold berdimensi rendah \mathcal{M} , dan grup transformasi G bertindak pada \mathcal{M} , maka terdapat representasi ϕ yang equivariant terhadap aksi G yang memproyeksikan \mathcal{M} ke koordinat intrinsiknya. Secara khusus, $\phi(g \cdot x) = T_g \phi(x)$ dengan T_g aksi linear pada ruang laten.*

Contoh 2.5.6 Dalam pengenalan wajah, meskipun setiap gambar memiliki ribuan piksel, variasi utamanya (arah pandangan, pencahayaan, ekspresi) dapat dijelaskan oleh beberapa koordinat laten berdimensi rendah. Fungsi ϕ bertugas memetakan gambar beresolusi tinggi tersebut ke ruang laten yang memuat koordinat intrinsik wajah yang sama terlepas dari variasi pose dan pencahayaan.

2.6. Pembelajaran Semi-Terawasi

Pembelajaran semi-terawasi (*semi-supervised learning*, SSL) adalah paradigma pembelajaran mesin yang memanfaatkan kombinasi data berlabel dalam jumlah kecil dan data tak berlabel dalam jumlah besar untuk meningkatkan kinerja model. Pendekatan ini sangat relevan dalam konteks praktis karena pelabelan data seringkali memerlukan biaya tinggi, waktu lama, dan keahlian khusus, sementara data tak berlabel dapat diperoleh dengan mudah dan murah. Dengan memanfaatkan struktur intrinsik dari data tak berlabel, SSL dapat menghasilkan model dengan generalisasi lebih baik dibandingkan pembelajaran terawasi murni yang hanya mengandalkan data berlabel terbatas.

2.6.1. Kerangka Dasar

Dalam banyak aplikasi pembelajaran mesin, memperoleh label memerlukan intervensi manusia yang mahal atau proses eksperimen yang memakan waktu. Sebagai contoh, dalam klasifikasi dokumen teks, pelabelan manual ribuan artikel memerlukan waktu dan keahlian domain yang signifikan. Sebaliknya, mengumpul-

an dokumen tak berlabel dari internet sangat mudah. SSL berupaya menjembatani kesenjangan ini dengan mengekstraksi informasi dari distribusi data tak berlabel untuk memperkaya proses pembelajaran.

Definisi 2.6.1 (Kerangka Pembelajaran Semi-Terawasi, (Chapelle, Schölkopf, & Zien, 2006)) Diberikan himpunan data berlabel $\mathcal{D}_\ell = \{(\mathbf{x}_i, y_i)\}_{i=1}^\ell$ dan himpunan data tak berlabel $\mathcal{D}_u = \{\mathbf{x}_j\}_{j=\ell+1}^n$, dengan $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^p$ adalah vektor fitur dan $y \in \mathcal{Y}$ adalah label (diskret untuk klasifikasi atau kontinu untuk regresi). Dalam konteks SSL, jumlah data tak berlabel umumnya jauh lebih besar daripada data berlabel, yaitu $u = n - \ell \gg \ell$.

Tujuan SSL adalah mempelajari fungsi prediksi $f : \mathcal{X} \rightarrow \mathcal{Y}$ yang memanfaatkan informasi dari kedua himpunan data. Formulasi umum masalah optimasi SSL adalah

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(f(\mathbf{x}_i), y_i)}_{\text{kerugian terawasi}} + \lambda \underbrace{\mathcal{R}(f; \mathcal{D}_u)}_{\text{regularisasi dari data tak berlabel}}, \quad (2.223)$$

dengan \mathcal{F} adalah kelas fungsi hipotesis, \mathcal{L} adalah fungsi kerugian terawasi, \mathcal{R} adalah term regularisasi yang mengekstrak struktur dari data tak berlabel, dan $\lambda > 0$ adalah parameter yang mengatur trade-off antara kedua komponen tersebut.

Contoh 2.6.2 Dalam klasifikasi dokumen, misalkan terdapat 100 dokumen berlabel tentang topik *politik* dan *olahraga*, serta 10,000 dokumen tak berlabel. SSL dapat memanfaatkan kemiripan antar dokumen tak berlabel untuk membentuk klaster yang koheren, sehingga model dapat belajar bahwa dokumen dengan kata-kata mirip cenderung memiliki label yang sama, meskipun label eksplisit tidak tersedia untuk sebagian besar dokumen.

Menurut Chapelle dkk. (2006), SSL dapat dikategorikan menjadi dua kerangka sebagai berikut.

- (a) Pembelajaran induktif, yaitu model mempelajari fungsi $f : \mathcal{X} \rightarrow \mathcal{Y}$ yang dapat digeneralisasikan ke data baru di luar himpunan pelatihan. Ini adalah

pendekatan umum yang digunakan dalam kebanyakan aplikasi modern.

- (b) Pembelajaran transduktif, yang tujuannya hanya untuk memprediksi label dari sejumlah data tak berlabel tertentu yang telah diketahui selama pelatihan, tanpa menggeneralisasi ke sampel baru.

Grandvalet and Bengio (2005) menambahkan bahwa SSL dapat dilihat sebagai jembatan antara pembelajaran induktif dan transduktif, karena ia memanfaatkan informasi struktur global dari semua data untuk memperbaiki estimasi pada titik tak berlabel yang diamati.

2.6.2. Asumsi Fundamental dalam Pembelajaran Semi-Terawasi

Keberhasilan SSL bergantung pada asumsi-asumsi tertentu tentang struktur data yang menghubungkan distribusi fitur $p(\mathbf{x})$ dengan distribusi label kondisional $p(y | \mathbf{x})$. Tanpa asumsi ini, data tak berlabel tidak memberikan informasi tambahan yang berguna (Chapelle dkk., 2006).

Definisi 2.6.3 (Asumsi Kehalusan (*Smoothness Assumption*), (Chapelle dkk., 2006)) *Jika dua titik $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ berada dalam daerah berdensitas tinggi dan berdekatan satu sama lain, maka label yang bersesuaian y_1 dan y_2 cenderung sama. Secara formal, fungsi f yang baik harus bersifat halus pada daerah berdensitas tinggi, yaitu*

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 \text{ kecil} \implies |f(\mathbf{x}_i) - f(\mathbf{x}_j)| \text{ kecil}, \quad (2.224)$$

untuk $\mathbf{x}_i, \mathbf{x}_j$ yang terletak pada daerah dengan $p(\mathbf{x})$ tinggi.

Contoh 2.6.4 Dalam pengenalan gambar wajah, dua gambar wajah yang hanya berbeda sedikit dalam pencahayaan atau rotasi kecil (jarak Euclidean kecil dalam ruang piksel) seharusnya memiliki identitas yang sama. Asumsi kehalusan menyatakan bahwa fungsi klasifikasi tidak boleh berubah drastis pada perubahan kecil seperti ini.

Definisi 2.6.5 (Asumsi Klaster (*Cluster Assumption*), (Chapelle dkk., 2006)) *Data cenderung membentuk klaster-klaster yang terpisah, dan titik-titik dalam klaster*

yang sama cenderung memiliki label yang sama. Batas keputusan yang baik seharusnya melewati daerah berdensitas rendah, bukan memotong klaster.

Secara matematis, jika terdapat jalur $\gamma : [0, 1] \rightarrow \mathcal{X}$ yang menghubungkan \mathbf{x}_i dengan \mathbf{x}_j sedemikian sehingga $p(\gamma(t))$ tinggi untuk semua $t \in [0, 1]$, maka $f(\mathbf{x}_i) \approx f(\mathbf{x}_j)$.

Contoh 2.6.6 Misalkan data terdiri dari dua klaster Gaussian yang terpisah jauh. Jika beberapa titik dari klaster pertama diberi label “positif” dan beberapa titik dari klaster kedua diberi label “negatif”, maka batas keputusan optimal seharusnya berada di antara kedua klaster (daerah berdensitas rendah), bukan memotong salah satu klaster.

Definisi 2.6.7 (Asumsi Manifold (Manifold Assumption), (Chapelle dkk., 2006))
Data berdimensi tinggi $\mathbf{x} \in \mathbb{R}^p$ sebenarnya terletak pada atau dekat dengan Manifold berdimensi rendah $\mathcal{M} \subset \mathbb{R}^p$. Fungsi f yang baik seharusnya bervariasi secara halus sepanjang struktur geometri Manifold \mathcal{M} ini, bukan sepanjang ruang ambien \mathbb{R}^p .

Asumsi ini memotivasi penggunaan regularisasi berbasis geometri, seperti Laplasian graf, untuk menekan variasi f sepanjang Manifold yang didefinisikan oleh data.

Contoh 2.6.8 Dalam *dataset* gambar wajah, meskipun setiap gambar memiliki ribuan piksel (dimensi tinggi), variasi sesungguhnya (pose, ekspresi, pencahayaan) berada pada Manifold berdimensi jauh lebih rendah. SSL yang memanfaatkan asumsi Manifold akan mempelajari representasi yang menghormati struktur geometri intrinsik ini.

2.6.3. Fungsi Kerugian untuk Pembelajaran Semi-Terawasi

Desain fungsi kerugian yang tepat merupakan aspek krusial dalam SSL karena menentukan bagaimana informasi dari data tak berlabel dimanfaatkan untuk memperbaiki model. Berbeda dengan pembelajaran terawasi yang hanya menggunakan kerugian pada data berlabel, SSL memerlukan komponen tambahan yang

mengeksktraksi informasi dari data tak berlabel (Chapelle dkk., 2006).

Definisi 2.6.9 (Struktur Umum Fungsi Kerugian SSL, (Chapelle dkk., 2006))

Fungsi kerugian total dalam SSL dapat didekomposisi menjadi tiga komponen:

$$\mathcal{L}_{total} = \mathcal{L}_{supervised} + \lambda_u \mathcal{L}_{unsupervised} + \lambda_r \mathcal{R}(\theta), \quad (2.225)$$

dengan:

- $\mathcal{L}_{supervised} = \frac{1}{\ell} \sum_{i=1}^{\ell} \ell(f_{\theta}(\mathbf{x}_i), y_i)$ adalah kerugian terawasi standar pada data berlabel,
- $\mathcal{L}_{unsupervised}$ adalah kerugian yang mengekstrak informasi dari data tak berlabel,
- $\mathcal{R}(\theta)$ adalah regularisasi pada parameter model,
- $\lambda_u, \lambda_r > 0$ adalah hyperparameter yang mengontrol kontribusi masing-masing komponen.

Komponen kunci yang membedakan berbagai metode SSL adalah desain $\mathcal{L}_{unsupervised}$. Beberapa pendekatan utama akan dijelaskan berikut ini.

Salah satu prinsip fundamental untuk memanfaatkan data tak berlabel adalah *entropy minimization*, yang didasarkan pada intuisi bahwa model yang baik seharusnya membuat prediksi yang percaya diri (*low entropy*) pada semua data, termasuk data tak berlabel (Grandvalet & Bengio, 2005).

Definisi 2.6.10 (Kerugian Entropi untuk SSL, (Grandvalet & Bengio, 2005))

Untuk klasifikasi multi-kelas dengan C kelas, diberikan model probabilistik yang menghasilkan distribusi prediksi $\hat{\mathbf{p}}(y | \mathbf{x}) = f_{\theta}(\mathbf{x}) \in \Delta^{C-1}$ (simplex probabilitas C -dimensi). Entropi kondisional dari prediksi adalah

$$H(y | \mathbf{x}; \theta) = - \sum_{c=1}^C \hat{p}_c(\mathbf{x}; \theta) \log \hat{p}_c(\mathbf{x}; \theta), \quad (2.226)$$

dengan $\hat{p}_c(\mathbf{x}; \theta)$ adalah probabilitas prediksi untuk kelas c .

Fungsi kerugian minimisasi entropi untuk data tak berlabel didefinisikan sebagai rata-rata entropi prediksi:

$$\mathcal{L}_{entropy} = \frac{1}{u} \sum_{j=\ell+1}^n H(y \mid \mathbf{x}_j; \theta) = -\frac{1}{u} \sum_{j=\ell+1}^n \sum_{c=1}^C \hat{p}_c(\mathbf{x}_j; \theta) \log \hat{p}_c(\mathbf{x}_j; \theta). \quad (2.227)$$

Meminimalkan $\mathcal{L}_{entropy}$ mendorong model untuk membuat prediksi yang lebih percaya diri (puncak distribusi lebih tajam) pada data tak berlabel.

Contoh 2.6.11 Untuk klasifikasi biner, misalkan model memprediksi probabilitas kelas positif $\hat{p}(\mathbf{x}) = 0.5$ untuk suatu titik tak berlabel. Entropi prediksi adalah

$$H = -[0.5 \log 0.5 + 0.5 \log 0.5] = -\log 0.5 \approx 0.693 \text{ (maksimum).}$$

Sebaliknya, jika model percaya diri dengan $\hat{p}(\mathbf{x}) = 0.95$, maka

$$H = -[0.95 \log 0.95 + 0.05 \log 0.05] \approx 0.198 \text{ (rendah).}$$

Minimisasi entropi mendorong model menuju prediksi percaya diri seperti kasus kedua.

Pendekatan lain yang populer adalah *consistency regularization*, yang didasarkan pada prinsip bahwa prediksi model seharusnya konsisten terhadap perturbasi kecil pada input atau model itu sendiri.

Definisi 2.6.12 (Kerugian Konsistensi, (Chapelle dkk., 2006)) Untuk augmentasi atau perturbasi \mathcal{T} (misalnya, *noise Gaussian*, *dropout*, *augmentasi data*), kerugian konsistensi didefinisikan sebagai

$$\mathcal{L}_{consistency} = \frac{1}{u} \sum_{j=\ell+1}^n \mathbb{E}_{\tau \sim \mathcal{T}} [\|\hat{\mathbf{p}}(\tau(\mathbf{x}_j); \theta) - \hat{\mathbf{p}}(\mathbf{x}_j; \theta)\|^2], \quad (2.228)$$

atau dalam bentuk praktis menggunakan sampling:

$$\mathcal{L}_{consistency} = \frac{1}{u} \sum_{j=\ell+1}^n \|\hat{\mathbf{p}}(\tau(\mathbf{x}_j); \theta) - \hat{\mathbf{p}}(\mathbf{x}_j; \theta)\|^2. \quad (2.229)$$

Kerugian ini mendorong model untuk menghasilkan prediksi yang sama untuk input yang telah diperturbasi dan input asli.

Contoh 2.6.13 Dalam klasifikasi gambar, augmentasi τ dapat berupa rotasi kecil, translasi, atau perubahan brightness. Untuk gambar wajah \mathbf{x}_j , jika model memprediksi identitas “Alice” dengan probabilitas 0.9, maka setelah rotasi 5 derajat $\tau(\mathbf{x}_j)$, model seharusnya tetap memprediksi “Alice” dengan probabilitas mendekati 0.9. Kerugian konsistensi menghukum deviasi antara kedua prediksi ini.

Teknik yang lebih sederhana namun efektif adalah *pseudo-labeling*, di mana model digunakan untuk membuat label sementara untuk data tak berlabel, yang kemudian diperlakukan sebagai label sebenarnya dalam iterasi pelatihan berikutnya.

Definisi 2.6.14 (Kerugian Pseudo-Labeling, (Chapelle dkk., 2006)) Untuk setiap titik tak berlabel \mathbf{x}_j , definisikan label sementara sebagai prediksi kelas dengan probabilitas tertinggi:

$$\tilde{y}_j = \arg \max_{c \in \{1, \dots, C\}} \hat{p}_c(\mathbf{x}_j; \theta). \quad (2.230)$$

Kerugian *pseudo-labeling* adalah:

$$\mathcal{L}_{pseudo} = \frac{1}{u} \sum_{j=\ell+1}^n \ell(f_\theta(\mathbf{x}_j), \tilde{y}_j), \quad (2.231)$$

dengan ℓ adalah fungsi kerugian standar (misalnya, *cross-entropy*).

Contoh 2.6.15 Misalkan model memprediksi distribusi probabilitas $[0.05, 0.15, 0.80]$ untuk tiga kelas pada titik tak berlabel \mathbf{x}_j . Pseudo-label adalah $\tilde{y}_j = 3$ (kelas ketiga). Pada iterasi pelatihan berikutnya, titik \mathbf{x}_j dengan pseudo-label 3 ditambahkan ke set pelatihan dan diperlakukan seolah-olah labelnya adalah 3.

Untuk data yang dapat direpresentasikan sebagai graf (atau ketika graf ke-

miripan dapat dikonstruksi), regularisasi berbasis Laplasian graf menyediakan cara alami untuk mengimplementasikan asumsi kehalusan.

Definisi 2.6.16 (Regularisasi Graf Laplacian, (Chapelle dkk., 2006)) Diberikan graf kemiripan dengan matriks bobot \mathbf{W} dan Laplasian $\mathbf{L} = \mathbf{D} - \mathbf{W}$, regularisasi graf untuk fungsi f didefinisikan sebagai

$$\mathcal{R}_{\text{graph}}(f) = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 = \mathbf{f}^\top \mathbf{L} \mathbf{f}, \quad (2.232)$$

dengan $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^\top$.

Untuk model neural yang menghasilkan representasi laten $\mathbf{h}_i = \phi_\theta(\mathbf{x}_i)$, regularisasi graf dapat diterapkan pada representasi:

$$\mathcal{L}_{\text{graph}} = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \|\mathbf{h}_i - \mathbf{h}_j\|^2. \quad (2.233)$$

Contoh 2.6.17 Untuk *dataset* dengan 5 titik dan graf kemiripan berbentuk jalur linear $1 - 2 - 3 - 4 - 5$ dengan bobot seragam $w_{ij} = 1$, jika representasi laten adalah $\mathbf{h} = [0, 0.2, 0.4, 0.6, 1]^\top$, maka

$$\begin{aligned} \mathcal{L}_{\text{graph}} &= \frac{1}{2} [|0 - 0.2|^2 + |0.2 - 0.4|^2 + |0.4 - 0.6|^2 + |0.6 - 1|^2] \\ &= \frac{1}{2} [0.04 + 0.04 + 0.04 + 0.16] = 0.14. \end{aligned}$$

Regularisasi ini mendorong representasi simpul yang berdekatan dalam graf untuk memiliki nilai yang mirip.

BAB III

PENGEMBANGAN PEMBOBOTAN BERBASIS JARINGAN SARAF GRAF UNTUK REGRESI SPASIAL-TEMPORAL

3.1. Kajian Literatur Terkait

Kajian mengenai regresi berbobot spasial dan spasial-temporal telah berkembang pesat dalam beberapa dekade terakhir. Evolusi model dapat ditelusuri mulai dari regresi klasik berbasis kernel, regresi berbobot geografis, hingga integrasi dengan pembelajaran mesin dan *neural network* serta *graph neural network*. Bagian ini memaparkan secara kronologis penelitian-penelitian terdahulu yang relevan dengan metodologi yang diusulkan.

3.1.1. *Geographically Weighted Artificial Neural Network* (GWANN)

Sebagai upaya mengintegrasikan *deep learning* ke dalam kerangka geografis, Hagenauer and Helbich (2022) mengusulkan *Geographically Weighted Artificial Neural Network* (GWANN). Pada GWANN, setiap lokasi memiliki model *artificial neural network* (ANN) lokal:

$$y_i = f(\mathbf{x}_i; \boldsymbol{\theta}(\mathbf{s}_i)) + \varepsilon_i, \quad (3.1)$$

dengan parameter $\boldsymbol{\theta}(\mathbf{s}_i)$ dipengaruhi oleh bobot kernel spasial. ANN lokal ini menggantikan model linear pada GWR, sehingga mampu menangkap hubungan non-linear antara kovariat dan respon. Kekurangannya adalah biaya komputasi yang tinggi karena ANN dilatih berulang pada tiap lokasi, serta ketidakstabilan parameter jika data lokal terbatas.

3.1.2. *Spatial and Attribute Neural Network Weighted Regression (SANNWR)*

Ni dkk. (2022) memperkenalkan *Spatial and Attribute Neural Network Weighted Regression* (SANNWR), yang memperluas GWR dengan bobot berbasis gabungan jarak spasial dan atribut. Estimasi regresi tetap menggunakan formulasi WLS, tetapi bobot w_{ij} dihitung melalui *neural network* yang mempelajari fungsi non-linear dari $(d_{ij}^{\text{spatial}}, d_{ij}^{\text{attribute}})$:

$$w_{ij} = \frac{\exp(\text{NN}(d_{ij}^{\text{spatial}}, d_{ij}^{\text{attribute}}))}{\sum_k \exp(\text{NN}(d_{ik}^{\text{spatial}}, d_{ik}^{\text{attribute}}))}. \quad (3.2)$$

Dengan demikian, bobot tidak lagi berupa fungsi kernel yang kaku, melainkan dipelajari secara adaptif. Inovasi ini menegaskan peran *deep learning* dalam merepresentasikan kedekatan spasial dan atribut secara simultan.

3.1.3. *Spatial Regression Graph Convolutional Neural Networks (SRGCNN)*

Zhu dkk. (2022) mengusulkan *Spatial Regression Graph Convolutional Neural Network* (SRGCNN), yang menggabungkan regresi spasial dengan konvolusi graf. Model ini merumuskan dependensi spasial dengan operator *graph convolutional network* (GCN), yaitu

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(\ell)} \mathbf{W}^{(\ell)}\right), \quad (3.3)$$

dan kemudian menghubungkannya dengan regresi linear autoregresif spasial (SAR), yaitu

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \rho \mathbf{W}\mathbf{y} + \boldsymbol{\varepsilon}, \quad (3.4)$$

sehingga efek lag spasial dan heterogenitas dapat ditangkap sekaligus. SRGCNN menandai transisi eksplisit penggunaan *graph neural network* dalam konteks regresi spasial. Sebagai catatan, SRGCNN tidak mengadopsi bobot yang dipelajari melalui GNN, melainkan menggunakan GNN untuk mengekstrak fitur spasial yang memiliki kemiripan dengan efek lag spasial pada model SAR.

3.1.4. *Spatio-Temporal Graph Convolutional Networks (ST-GCN)*

Dalam domain peramalan deret waktu spatiotemporal, Yu, Yin, and Zhu (2018) mengusulkan *Spatio-Temporal GCN* (ST-GCN). Model ini mengintegrasikan konvolusi graf spasial dengan konvolusi temporal berbasis deret:

$$\mathbf{H}^{(\ell+1)} = \sigma(\text{ConvTemporal}(\tilde{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)})). \quad (3.5)$$

ST-GCN dan variannya (ASTGCN, T-GCN) banyak digunakan untuk memprediksi lalu lintas, konsumsi energi, dan dinamika jaringan kompleks. Kontribusinya adalah menunjukkan bahwa *graph neural network* dapat diperluas secara alami ke dimensi waktu.

3.1.5. *Geographically Neural Network Weighted Regression (GNNWR)*

Salah satu perkembangan mutakhir dalam literatur regresi spasial adalah pengenalan *Geographically Neural Network Weighted Regression* (GNNWR) oleh Yin dkk. (2024). Model ini berangkat dari keterbatasan *Geographically Weighted Regression* (GWR) yang menggunakan fungsi *kernel* parametrik untuk menghitung bobot spasial. Pada GWR, estimator koefisien lokal diberikan oleh

$$\hat{\beta}(u_i, v_i) = (\mathbf{X}^\top \mathbf{W}(u_i, v_i) \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}(u_i, v_i) \mathbf{y}, \quad (3.6)$$

dengan $\mathbf{W}(u_i, v_i)$ matriks bobot *kernel* spasial berbasis jarak Euclidean antara titik observasi i dan titik lainnya. Namun, pemilihan *kernel* yang kaku sering gagal menangkap heterogenitas non-linear. Untuk itu, GNNWR mengganti $\mathbf{W}(u_i, v_i)$ dengan bobot yang dipelajari oleh *spatial weighted neural network* (SWNN). Secara umum, model GNNWR memandang koefisien lokal sebagai skala dari koefisien global,

$$\beta_k^{\text{Lokal}}(u_i, v_i) = w_k(u_i, v_i) \beta_k^{\text{Global}}, \quad \sum_{k=1}^p w_k(u_i, v_i) = 1, \quad w_k \geq 0, \quad (3.7)$$

dengan β_k^{Global} diperoleh dari model OLS *baseline*. Hubungan ini memastikan bahwa koefisien global dapat diinterpretasikan sebagai rata-rata tertimbang dari koefisien lokal.

Bobot $w_k(u_i, v_i)$ dipelajari menggunakan *neural network* yang menerima input berupa jarak spasial antar unit:

$$\mathbf{W}(u_i, v_i) = \text{SWNN}(d_{i1}^S, d_{i2}^S, \dots, d_{in}^S; \boldsymbol{\theta}), \quad (3.8)$$

dengan d_{ij}^S jarak spasial antara unit i dan j , serta $\boldsymbol{\theta}$ parameter jaringan saraf. Oleh karena itu, prediksi \hat{y}_i dapat dituliskan sebagai

$$\hat{y}_i = \mathbf{x}_i^\top \mathbf{W}(u_i, v_i) (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \quad i = 1, \dots, n. \quad (3.9)$$

Persamaan (3.9) dapat dipadatkan menjadi bentuk matriks

$$\hat{\mathbf{y}} = \mathbf{S} \mathbf{y}, \quad (3.10)$$

dengan \mathbf{S} bergantung pada bobot $\mathbf{W}(u_i, v_i)$ yang dilatih melalui *neural network*.

Yin dkk. (2024) kemudian memperluas kerangka ini menjadi *Geographically and Temporally Neural Network Weighted Regression* (GTNNWR), yang memasukkan dimensi waktu ke dalam pembobotan. Estimator GTNNWR ditulis sebagai

$$\hat{y}_i = \sum_{k=1}^p w_k(u_i, v_i, t) \beta_k^{\text{Global}} x_{ik} + \varepsilon_i, \quad (3.11)$$

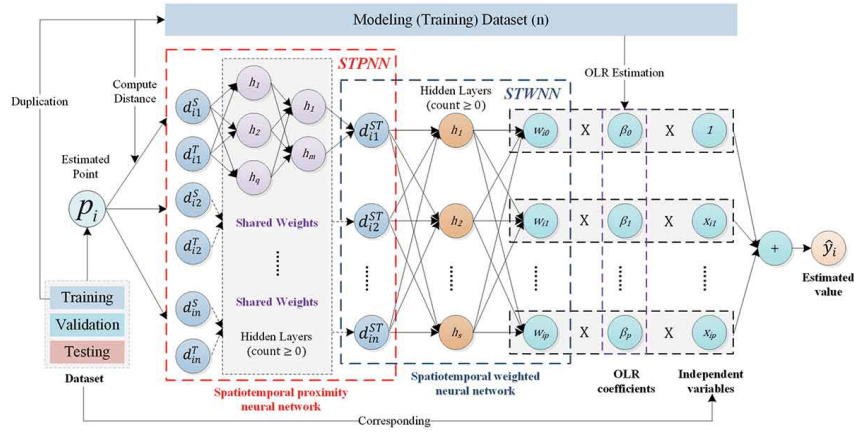
dengan bobot spasiotemporal

$$\mathbf{W}(u_i, v_i, t) = \text{STWNN}(d_{i1}^{ST}, d_{i2}^{ST}, \dots, d_{in}^{ST}; \boldsymbol{\theta}_1), \quad (3.12)$$

di mana jarak spasiotemporal d_{ij}^{ST} diperoleh melalui *spatiotemporal proximity neural network* (STPNN), yaitu

$$d_{ij}^{ST} = \text{STPNN}(d_{ij}^S, d_{ij}^T; \boldsymbol{\theta}_2). \quad (3.13)$$

Dengan demikian, GTNNWR melibatkan dua tahap pembelajaran, yaitu STPNN untuk menyintesis jarak spasiotemporal dari komponen spasial dan temporal dan STWNN untuk mengubah jarak tersebut menjadi bobot regresi. Diagram kerangka kerja GTNNWR ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Kerangka Kerja GTNNWR (Yin dkk., 2024)

Sebagaimana pada GWR, prediksi \hat{y} dalam GNNWR dan GTNNWR tetap dapat dipandang melalui matriks Sy . Akan tetapi, W kini dihasilkan secara endogen melalui *neural network*. Fungsi kerugian yang umum digunakan adalah *mean squared error* (MSE), yaitu

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (3.14)$$

serta kriteria informasi seperti *Akaike Information Criterion* (AIC) dan AICc untuk menghindari *overfitting*, yaitu

$$AIC_c = 2k - 2\ln(L) + \frac{2k(k+1)}{n-k-1}, \quad (3.15)$$

dengan k jumlah parameter efektif dan L adalah *likelihood* model.

Metode GNNWR dan GTNNWR telah diaplikasikan pada berbagai domain, antara lain pemodelan kualitas udara (PM_{2.5} dan AOD), pemetaan nutrisi laut (*dissolved silicate*), serta estimasi emisi karbon perkotaan. Studi-studi tersebut menunjukkan bahwa dengan menggantikan kernel spasial konvensional dengan *ne-*

ural network, model mampu menangkap heterogenitas spasiotemporal yang lebih kompleks, sekaligus mempertahankan interpretabilitas koefisien lokal sebagaimana pada GWR.

3.2. Kerangka Metodologis

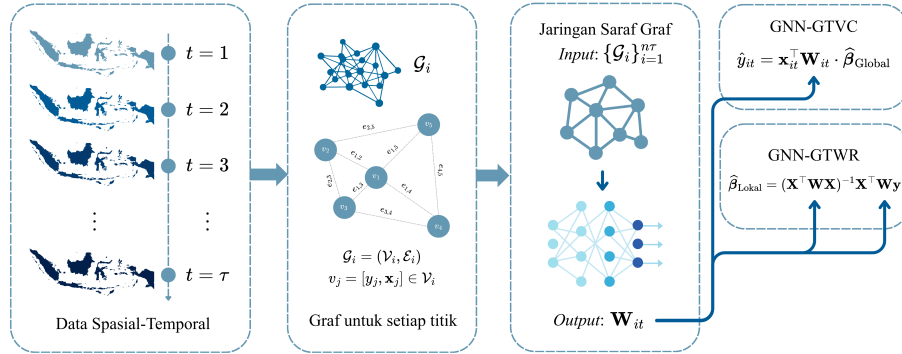
Bagian ini menjelaskan kerangka metodologis yang digunakan dalam pengembangan dua model utama, yaitu *Graph Neural Network-Geographically and Temporally Varying Coefficient* (GNN-GTVC) dan *Graph Neural Network-Geographically and Temporally Weighted Regression* (GNN-GTWR). Kedua model dikembangkan untuk menangkap heterogenitas spasial dan temporal yang kompleks dengan mengintegrasikan kemampuan representasi *Graph Neural Network* (GNN) ke dalam kerangka regresi spasiotemporal berbobot.

Keterbatasan utama pendekatan kernel pada GWR/GTWR (Fotheringham dkk., 2002; Huang dkk., 2010) adalah penggunaan fungsi jarak parametrik yang kaku, sehingga sering gagal menangkap heterogenitas spasio-temporal yang non-linear. Literatur mutakhir seperti GWANN, SANNWR, dan SRGCNN menunjukkan manfaat *deep learning* untuk mempelajari kedekatan efektif yang adaptif. Namun, terdapat celah metodologis penting, yaitu GNNWR/GTNNWR (sebagaimana dipaparkan dalam Yin dkk. 2024) tidak melakukan estimasi koefisien lokal melalui WLS sebagaimana GWR/GTWR klasik, melainkan mengadopsi *skema multiplicative* yang pada hakikatnya adalah *varying-coefficient model* (VCM) Hastie and Tibshirani (1993). Di sisi lain, belum ada kerangka yang secara sistematis menggabungkan dua paradigma tersebut, yaitu bobot dipelajari secara multiplikatif dalam VCM dan bobot yang dipelajari dalam rumus estimator WLS dalam satu metodologi yang konsisten, transparan, dan dapat dianalisis.

3.2.1. Kerangka Umum dan Alur Metodologi

Secara umum, rancangan metodologi penelitian ini mengikuti alur yang ditunjukkan pada Gambar 3.2. *Pipeline* model dibangun untuk memetakan data panel spasiotemporal ke dalam bentuk graf dinamis, mengekstraksi representasi spasi-

al dan temporal melalui GNN, kemudian mengonversi hasil representasi tersebut menjadi bobot adaptif yang digunakan dalam proses estimasi koefisien regresi.



Gambar 3.2 Alur metodologi GNN-GTVC dan GNN-GTWR. (Sumber: Dokumen penulis)

Secara garis besar, kerangka metodologi ini terdiri atas lima tahapan utama yang saling berkaitan:

- (a) **Konstruksi data dan graf spasiotemporal.** Data berbentuk panel dengan N unit spasial (misal provinsi) dan T periode waktu. Relasi spasial dibentuk menggunakan k -nearest neighbors (k-NN) berdasarkan jarak geografis antarunit, sedangkan relasi temporal menghubungkan setiap simpul (i, t) dengan $(i, t - 1)$ dan $(i, t + 1)$. Kombinasi kedua relasi tersebut menghasilkan graf spasiotemporal dinamis $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ dengan $\mathcal{V} = \{(i, t)\}$ sebagai himpunan simpul.
- (b) **Pembelajaran representasi graf melalui GNN.** Setiap simpul (i, t) memiliki vektor fitur $\mathbf{x}_{i,t}$ dan target $y_{i,t}$. Melalui operasi konvolusi graf, GNN menghasilkan representasi laten (*embedding*) $\mathbf{h}_{i,t}$ yang merepresentasikan hubungan spasial dan temporal secara adaptif. Tiga arsitektur yang digunakan adalah GCN, GAT, dan GraphSAGE.
- (c) **Pemetaan representasi ke bobot regresi.** Representasi laten $\mathbf{h}_{i,t}$ diproyeksikan menjadi bobot w , baik dalam bentuk bobot antar-koefisien (pada GNN-GTVC) maupun bobot antar-tetangga (pada GNN-GTWR). Pemetaan ini dirancang agar bobot yang dihasilkan bersifat nonnegatif, ternormalisasi, dan

dapat diinterpretasikan secara probabilistik.

- (d) **Regresi spasiotemporal dengan koefisien bervariasi atau terboboti.** Bobot hasil pembelajaran GNN digunakan dalam dua kerangka regresi. Pada GNN-GTVC, bobot mengatur kontribusi relatif setiap kovariat terhadap respon di setiap lokasi dan waktu, menghasilkan model dengan koefisien lokal yang bervariasi. Pada GNN-GTWR, bobot mengatur pengaruh observasi lain dalam estimasi parameter lokal melalui pendekatan *weighted least squares* (WLS).
- (e) **Pelatihan dan optimisasi model.** Parameter jaringan, baik pada bagian GNN maupun regresi, dilatih dengan meminimalkan fungsi kerugian total yang menggabungkan komponen terawasi (*supervised loss*) dan regularisasi spasiotemporal. Tahap ini juga dapat diperluas dengan mekanisme pembelajaran semi-terawasi (*semi-supervised learning*, SSL) untuk memanfaatkan data tak berlabel.

3.2.2. Alur Pembelajaran GNN-GTVC dan GNN-GTWR

Kedua model memiliki struktur pembelajaran yang serupa, tetapi berbeda dalam penerapan bobot hasil GNN. Pada GNN-GTVC, pembobotan dilakukan pada tingkat koefisien, sedangkan pada GNN-GTWR, pembobotan dilakukan pada tingkat observasi.

Pada GNN-GTVC, GNN terlebih dahulu mempelajari representasi spasiotemporal $\mathbf{h}_{i,t}$ untuk setiap simpul (i, t) . Representasi tersebut kemudian diproyeksikan ke dalam simplex bobot $\mathbf{w}_{i,t} \in \Delta^{p-1}$ menggunakan fungsi softmax berbasis *dot-product*, *cosine similarity*, atau *multilayer perceptron* (MLP). Bobot $\mathbf{w}_{i,t}$ ini mengatur proporsi kontribusi masing-masing kovariat $x_{ik,t}$ terhadap respon $y_{i,t}$ melalui relasi

$$y_{i,t} = \sum_{k=1}^p w_{i,t}^{(k)} \beta_k^{\text{global}} x_{ik,t} + \varepsilon_{i,t}. \quad (3.16)$$

Koefisien global β_k^{global} diperoleh melalui estimasi OLS, sedangkan bobot $\mathbf{w}_{i,t}$ dilatih melalui propagasi balik berdasarkan fungsi kerugian model. Koefisien OLS

dapat diganti dengan koefisien dari model regresi lain, seperti Ridge atau Lasso, sesuai kebutuhan.

Pada GNN-GTWR, representasi spasiotemporal $\mathbf{h}_{i,t}$ diperoleh melalui GNN dan dibandingkan dengan representasi simpul lain (misalnya tetangga terdekat) untuk menghasilkan bobot lokal $\tilde{w}_{(i,t) \rightarrow (j,s)}$, sebagai contoh menggunakan fungsi softmax berbasis *dot-product* yaitu

$$\tilde{w}_{(i,t) \rightarrow (j,s)} = \frac{\exp(\mathbf{h}_{i,t}^\top \mathbf{h}_{j,s} / \tau)}{\sum_{(k,r) \in \mathcal{N}(i,t)} \exp(\mathbf{h}_{i,t}^\top \mathbf{h}_{k,r} / \tau)}. \quad (3.17)$$

Bobot ini kemudian disusun ke dalam matriks diagonal $\mathbf{W}_{i,t}$ untuk memperkirakan koefisien lokal melalui metode WLS:

$$\hat{\beta}_{i,t} = (\mathbf{X}^\top \mathbf{W}_{i,t} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_{i,t} \mathbf{y}. \quad (3.18)$$

Prediksi lokal diperoleh dengan $\hat{y}_{i,t} = \mathbf{x}_{i,t}^\top \hat{\beta}_{i,t}$, dan gradien kerugian dipropagasikan kembali ke parameter GNN untuk memperbarui bobot jaringan.

3.2.3. Perbandingan Konseptual Awal

Kedua model berangkat dari prinsip yang sama, yaitu memperkaya model regresi spasiotemporal dengan pembobotan adaptif berbasis graf. Perbedaannya terletak pada tingkat penerapan bobot yang dihasilkan. GNN-GTVC mempelajari bobot antar-koefisien (*coefficient-level weighting*), sehingga lebih sesuai untuk menelusuri dinamika kontribusi variabel terhadap respon pada setiap lokasi dan waktu. Sebaliknya, GNN-GTWR mempelajari bobot antar-observasi (*sample-level weighting*), yang lebih dekat dengan pendekatan klasik GWR atau GTWR namun dengan mekanisme pembobotan nonparametrik yang adaptif. Kedua pendekatan tersebut bersifat saling melengkapi. Model GTVC menekankan interpretabilitas koefisien lokal, sedangkan GTWR menekankan stabilitas estimasi dan kemampuan generalisasi spasial. Bagian selanjutnya akan membahas formulasi matematis dan komputasional masing-masing model secara lebih rinci, dimulai dari GNN-GTVC pada Bagian 3.3. dan dilanjutkan dengan GNN-GTWR pada Bagian 3.4..

3.3. Regresi Spasial-Temporal Berkoeffisien Bervariasi berbasis GNN (GNN-GTVC)

Bagian ini menguraikan secara rinci formulasi matematis, pembentukan bobot koefisien lokal, serta proses estimasi dan interpretasi pada model GNN-GTVC. Model ini dikembangkan sebagai perluasan dari konsep *varying coefficient model* yang parameter lokalnya dipelajari secara adaptif melalui jaringan saraf graf. Pendekatan ini memungkinkan setiap lokasi dan waktu memiliki kombinasi kovariat yang berbeda-beda, dengan pembobotan yang diatur secara endogen oleh representasi spasiotemporal.

3.3.1. Formulasi Matematis

Misalkan tersedia himpunan observasi spasiotemporal $\mathcal{V} = \{(i, t) \mid i = 1, \dots, N; t = 1, \dots, T\}$ dengan respon $y_{i,t}$ dan kovariat $\mathbf{x}_{i,t} = (x_{i1,t}, x_{i2,t}, \dots, x_{ip,t})^\top$. Model GNN-GTVC dituliskan dalam bentuk umum

$$y_{i,t} = \sum_{k=1}^p \beta_k(z_{i,t}) x_{ik,t} + \varepsilon_{i,t}, \quad z_{i,t} = (u_i, v_i, t), \quad (3.19)$$

dengan $\beta_k(z_{i,t})$ merupakan koefisien yang bervariasi secara spasiotemporal dan $\varepsilon_{i,t}$ adalah komponen galat dengan $\mathbb{E}[\varepsilon_{i,t} \mid X] = 0$.

Untuk menghubungkan model ini dengan jaringan graf, koefisien lokal $\beta_k(z_{i,t})$ diparametrisasi sebagai fungsi dari koefisien global β_k^{global} dan bobot adaptif $w_k(z_{i,t})$:

$$\beta_k(z_{i,t}) = w_k(z_{i,t}) \beta_k^{\text{global}}, \quad w_k(z_{i,t}) \geq 0, \quad \sum_{k=1}^p w_k(z_{i,t}) = 1. \quad (3.20)$$

Dengan substitusi persamaan di atas ke dalam model umum (3.19) diperoleh bentuk regresi

$$y_{i,t} = \sum_{k=1}^p w_k(z_{i,t}) \beta_k^{\text{global}} x_{ik,t} + \varepsilon_{i,t}. \quad (3.21)$$

Koefisien global β_k^{global} dapat diestimasi terlebih dahulu menggunakan regresi OLS pada seluruh data, kemudian bobot $w_k(z_{i,t})$ dilatih menggunakan jaringan saraf graf

agar fungsi pembobotan mampu menangkap pola variasi lokal yang kompleks.

3.3.2. Pembentukan Bobot Koefisien Lokal melalui Representasi Graf

Pada tahap ini, setiap simpul (i, t) dalam graf spasiotemporal memiliki representasi laten $\mathbf{h}_{i,t} \in \mathbb{R}^d$ yang dihasilkan oleh jaringan GNN. Representasi tersebut kemudian diproyeksikan ke dalam *simplex* berdimensi p untuk menghasilkan bobot antar-koefisien $\mathbf{w}_{i,t} = (w_{i,t}^{(1)}, \dots, w_{i,t}^{(p)})^\top$. Beberapa bentuk pemetaan yang dapat digunakan antara lain

$$w_{i,t}^{(k)} = \frac{\exp(\mathbf{h}_{i,t}^\top \mathbf{v}_k / \tau)}{\sum_{j=1}^p \exp(\mathbf{h}_{i,t}^\top \mathbf{v}_j / \tau)}, \quad (3.22)$$

$$w_{i,t}^{(k)} = \frac{\exp(-\|\mathbf{h}_{i,t} - \mathbf{v}_k\|^2 / (2\sigma^2))}{\sum_{j=1}^p \exp(-\|\mathbf{h}_{i,t} - \mathbf{v}_j\|^2 / (2\sigma^2))}, \quad (3.23)$$

masing-masing merepresentasikan proyeksi berbasis *dot-product* dan kernel Gaussian. Vektor \mathbf{v}_k merupakan representasi yang dipelajari untuk kovariat ke- k , sedangkan τ dan σ adalah parameter penghalus yang mengatur ketajaman distribusi bobot.

Proses ini memastikan bahwa seluruh bobot bernilai nonnegatif dan ternormalisasi, sehingga setiap $\mathbf{w}_{i,t}$ dapat ditafsirkan sebagai distribusi probabilitas atas kovariat. Dengan demikian, setiap lokasi-waktu memiliki kombinasi unik dari kovariat yang berkontribusi terhadap respon, bergantung pada struktur spasiotemporal yang dipelajari GNN.

3.3.3. Estimasi Koefisien Global dan Prediksi

Koefisien global $\boldsymbol{\beta}^{\text{global}} = (\beta_1^{\text{global}}, \dots, \beta_p^{\text{global}})^\top$ diestimasi dengan meminimalkan jumlah kuadrat kesalahan secara keseluruhan:

$$\hat{\boldsymbol{\beta}}^{\text{global}} = \arg \min_{\boldsymbol{\beta}} \sum_{i,t} (y_{i,t} - \mathbf{x}_{i,t}^\top \mathbf{W}_{i,t} \boldsymbol{\beta})^2, \quad (3.24)$$

dengan $\mathbf{W}_{i,t} = \text{diag}(w_{i,t}^{(1)}, \dots, w_{i,t}^{(p)})$. Prediksi pada titik (i, t) diperoleh melalui

$$\hat{y}_{i,t} = \sum_{k=1}^p w_{i,t}^{(k)} \hat{\beta}_k^{\text{global}} x_{ik,t}. \quad (3.25)$$

Seluruh parameter jaringan $\Theta = \{\mathbf{v}_k, \text{parameter GNN}\}$ dioptimalkan secara bersamaan menggunakan propagasi balik dengan meminimalkan fungsi kerugian total yang akan dibahas pada Bagian 4.7.

3.4. Regresi Spasial-Temporal Terboboti berbasis GNN (GNN-GTWR)

Bagian ini menjelaskan formulasi matematis, pembentukan bobot antartetangga, serta proses estimasi dan prediksi pada model GNN-GTWR. Model ini dikembangkan sebagai perluasan dari *Geographically and Temporally Weighted Regression* (GTWR) klasik dengan pembobotan yang dipelajari secara adaptif melalui jaringan saraf graf. Pendekatan ini memungkinkan bobot spasiotemporal tidak lagi ditentukan oleh fungsi kernel parametrik, melainkan dipelajari langsung dari struktur relasi antarsampel menggunakan GNN.

3.4.1. Formulasi Estimator WLS pada Konteks Graf

Misalkan tersedia pasangan observasi $(\mathbf{x}_{i,t}, y_{i,t})$ dengan $i = 1, \dots, N$ dan $t = 1, \dots, T$. Pada model GTWR klasik, koefisien lokal di lokasi-waktu (i, t) diperoleh melalui estimasi *weighted least squares* (WLS) sebagai berikut:

$$\hat{\beta}_{i,t} = (\mathbf{X}^\top \mathbf{W}_{i,t} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_{i,t} \mathbf{y}, \quad (3.26)$$

dengan $\mathbf{W}_{i,t} = \text{diag}(w_{(i,t) \rightarrow (j,s)})$ merupakan matriks bobot spasiotemporal yang mencerminkan kedekatan antara titik (i, t) dan seluruh observasi (j, s) . Dalam GNN-GTWR, matriks bobot ini tidak lagi ditentukan oleh jarak Euclidean atau *kernel* tetap, tetapi dihasilkan secara endogen melalui mekanisme pembelajaran graf.

3.4.2. Pembobotan Antartetangga dengan Representasi GNN

Pembobotan pada GNN-GTWR dibangun dari representasi laten setiap simpul (i, t) yang diperoleh melalui GNN. Representasi $\mathbf{h}_{i,t} \in \mathbb{R}^d$ berfungsi sebagai penyandi (encoding) kedekatan spasial dan temporal. Bobot antar-simpul kemudian ditentukan dengan mengukur kesamaan antarrepresentasi, yang dinormalisasi agar bersifat stokastik baris:

$$\tilde{w}_{(i,t) \rightarrow (j,s)} = \frac{\exp(\mathbf{h}_{i,t}^\top \mathbf{h}_{j,s} / \tau)}{\sum_{(k,r) \in \mathcal{N}(i,t)} \exp(\mathbf{h}_{i,t}^\top \mathbf{h}_{k,r} / \tau)}, \quad (3.27)$$

dengan $\tau > 0$ sebagai parameter suhu (*temperature*) yang mengatur ketajaman distribusi bobot. Himpunan $\mathcal{N}(i, t)$ menunjukkan tetangga-tetangga terdekat dari simpul (i, t) yang ditentukan melalui kriteria spasial dan temporal. Prosedur ini menghasilkan matriks bobot $\mathbf{W}_{i,t}$ yang jarang (*sparse*) dan lokal, sehingga komputasi estimasi tetap efisien.

3.5. Arsitektur Jaringan Graf untuk Pembelajaran Bobot

Dalam penelitian ini digunakan tiga arsitektur utama jaringan saraf graf, yaitu *Graph Convolutional Network* (GCN), *Graph Attention Network* (GAT), dan GraphSAGE. Ketiganya berperan dalam mengekstraksi representasi laten spasio-temporal dari data panel yang telah dipetakan ke graf.

GCN merupakan arsitektur dasar yang memperbarui representasi setiap simpul dengan merata-ratakan informasi dari tetangga, menggunakan propagasi

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \mathbf{H}^{(\ell)} \mathbf{W}^{(\ell)}\right), \quad (3.28)$$

dengan $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ dan $\tilde{\mathbf{D}}$ matriks derajat diagonal. Arsitektur ini sederhana dan efisien, namun kurang fleksibel dalam membedakan pengaruh tiap tetangga.

GAT memperkenalkan mekanisme perhatian (*attention*) sehingga setiap simpul dapat memberi bobot berbeda pada tetangganya. Koefisien perhatian dihitung

sebagai

$$\alpha_{ij}^{(\ell)} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i^{(\ell)} \parallel \mathbf{W}\mathbf{h}_j^{(\ell)}]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i^{(\ell)} \parallel \mathbf{W}\mathbf{h}_k^{(\ell)}]\right)\right)}, \quad (3.29)$$

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(\ell)} \mathbf{W}\mathbf{h}_j^{(\ell)}\right). \quad (3.30)$$

Dengan demikian, GAT lebih adaptif dalam menangkap ketergantungan dinamis antar simpul.

Sementara itu, GraphSAGE mengagregasi informasi tetangga menggunakan fungsi agregasi terparametrisasi, misalnya rata-rata, *pooling*, atau LSTM, dan memperbarui *embedding* dengan

$$\mathbf{h}_i^{(\ell+1)} = \phi\left(\mathbf{h}_i^{(\ell)} \parallel \text{AGGREGATE}(\{\mathbf{h}_j^{(\ell)} : j \in \mathcal{N}(i)\})\right), \quad (3.31)$$

dengan ϕ *multilayer perceptron* kecil. GraphSAGE cocok untuk generalisasi ke simpul atau waktu baru.

Ketiga arsitektur ini menghasilkan *embedding* $\mathbf{h}_{i,t}$ yang kemudian diproyeksikan menjadi bobot regresi adaptif sesuai kebutuhan model.

3.6. Pilihan Pemetaan Representasi ke Bobot

Pemetaan *embedding* hasil GNN ke bobot regresi dapat dilakukan dengan beberapa metode utama berikut, beserta keunggulan dan interpretasinya:

(a) ***Dot-product***:

$$w_{i,t}^{(k)} = \frac{\exp(\mathbf{h}_{i,t}^\top \mathbf{v}_k / \tau)}{\sum_{j=1}^p \exp(\mathbf{h}_{i,t}^\top \mathbf{v}_j / \tau)} \quad (3.32)$$

Keunggulan utama pendekatan ini terletak pada kesederhanaan dan efisiensi komputasi, karena hanya memerlukan operasi perkalian vektor dan normalisasi softmax. Secara konseptual, metode ini merepresentasikan kemiripan linear antara *embedding node* $\mathbf{h}_{i,t}$ dan vektor representasi koefisien \mathbf{v}_k , sehingga bobot yang dihasilkan akan lebih besar apabila kedua vektor tersebut

memiliki arah yang sejalan dalam ruang fitur.

(b) ***Cosine similarity***:

$$w_{i,t}^{(k)} = \frac{\exp\left(\frac{\mathbf{h}_{i,t}^\top \mathbf{v}_k}{\|\mathbf{h}_{i,t}\| \|\mathbf{v}_k\| \tau}\right)}{\sum_{j=1}^p \exp\left(\frac{\mathbf{h}_{i,t}^\top \mathbf{v}_j}{\|\mathbf{h}_{i,t}\| \|\mathbf{v}_j\| \tau}\right)} \quad (3.33)$$

Pendekatan *cosine similarity* memiliki keunggulan invariansi terhadap skala vektor, sehingga hanya mempertimbangkan arah relatif antara *embedding* dan vektor koefisien. Hal ini menjadikan metode ini lebih *robust* terhadap variasi *magnitude embedding*, dan secara interpretatif bobot yang dihasilkan mencerminkan kedekatan arah (sudut) antara dua vektor dalam ruang fitur.

(c) **Gaussian (RBF) kernel**:

$$w_{i,t}^{(k)} = \frac{\exp(-\|\mathbf{h}_{i,t} - \mathbf{v}_k\|^2 / (2\sigma^2))}{\sum_{j=1}^p \exp(-\|\mathbf{h}_{i,t} - \mathbf{v}_j\|^2 / (2\sigma^2))} \quad (3.34)$$

Keunggulan utama kernel Gaussian terletak pada kemampuannya menangkap hubungan nonlinear antara *embedding* dan vektor koefisien, karena bobot ditentukan berdasarkan jarak Euclidean di ruang fitur. Secara interpretatif, bobot yang tinggi diberikan pada pasangan *embedding* dan vektor koefisien yang letaknya berdekatan, sehingga metode ini sensitif terhadap kemiripan lokal dalam representasi laten.

(d) **MLP (*Multilayer Perceptron*)**:

$$\mathbf{w}_{i,t} = \text{softmax}(\text{MLP}(\mathbf{h}_{i,t})) \quad (3.35)$$

Penggunaan *multilayer perceptron* (MLP) sebagai pemetaan dari *embedding* ke bobot memungkinkan model mempelajari fungsi nonlinear yang sangat kompleks tanpa asumsi eksplisit mengenai bentuk hubungan antara *embedding* dan bobot. Keunggulan metode ini terletak pada kapasitas representasi yang tinggi, sehingga mampu mengakomodasi pola interaksi yang rumit dalam data spasiotemporal.

3.6.1. Fungsi Objektif dan Proses Pelatihan

Setelah *embedding* dipetakan menjadi bobot \mathbf{W} , tahap selanjutnya adalah pelatihan model melalui optimisasi fungsi objektif. Fungsi objektif ini dirancang untuk mengukur deviasi antara nilai aktual y dan prediksi \hat{y} yang dihasilkan oleh model GTVC maupun GTWR.

Fungsi objektif utama yang digunakan adalah *mean squared error (MSE)*, yang didefinisikan sebagai

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (3.36)$$

dengan n menyatakan jumlah observasi, y_i merupakan nilai aktual, dan \hat{y}_i adalah prediksi yang dihasilkan model. Formulasi (3.36) bersifat umum dan dapat diterapkan baik pada GTVC maupun GTWR.

Sebagai alternatif, beberapa literatur mengadopsi kriteria informasi seperti *Akaike Information Criterion* terkoreksi (AIC_c) untuk mengendalikan kompleksitas model:

$$AIC_c = 2k - 2 \ln(L) + \frac{2k(k+1)}{n-k-1}, \quad (3.37)$$

dengan k menyatakan jumlah parameter efektif dan L adalah nilai *likelihood* model. Kriteria ini relevan terutama untuk mencegah *overfitting* pada model dengan banyak parameter bobot yang dipelajari.

Untuk meningkatkan stabilitas, fungsi objektif dapat dilengkapi dengan komponen regularisasi berikut:

(a) **Entropi bobot:**

$$\mathcal{R}_{\text{entropy}} = - \sum_i \sum_j \mathbf{w}_{ij} \log \mathbf{w}_{ij}, \quad (3.38)$$

yang berfungsi untuk mencegah distribusi bobot menjadi terlalu tajam (*over-confident*).

(b) **Regularisasi kelancaran spasiotemporal:**

$$\mathcal{R}_{\text{smooth}} = \sum_{(i,t) \sim (j,s)} \mathbf{w}_{(i,t),(j,s)} \|\boldsymbol{\beta}_{i,t} - \boldsymbol{\beta}_{j,s}\|^2, \quad (3.39)$$

yang mendorong agar koefisien pada lokasi dan waktu yang berdekatan tidak berbeda secara ekstrem, sesuai prinsip kontinuitas spasiotemporal.

Dengan demikian, fungsi objektif total dapat dituliskan sebagai

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_1 \mathcal{R}_{\text{entropy}} + \lambda_2 \mathcal{R}_{\text{smooth}}, \quad (3.40)$$

dengan $\lambda_1, \lambda_2 \geq 0$ sebagai parameter regularisasi.

Proses pelatihan dilakukan menggunakan algoritma *stochastic gradient descent* (SGD). Dua algoritma populer yang digunakan adalah:

(a) ***Stochastic Gradient Descent* (SGD):**

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}), \quad (3.41)$$

dengan η sebagai laju pembelajaran (*learning rate*) dan $\boldsymbol{\theta}$ parameter model.

(b) ***Adam optimizer*:** varian adaptif dari *SGD* yang memperbarui parameter dengan estimasi momentum gradien pertama dan kedua:

$$\mathbf{m}^{(t+1)} = \beta_1 \mathbf{m}^{(t)} + (1 - \beta_1) \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}), \quad (3.42)$$

$$\mathbf{v}^{(t+1)} = \beta_2 \mathbf{v}^{(t)} + (1 - \beta_2) (\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}))^2, \quad (3.43)$$

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \frac{\mathbf{m}^{(t+1)} / (1 - \beta_1^{t+1})}{\sqrt{\mathbf{v}^{(t+1)} / (1 - \beta_2^{t+1})} + \epsilon}. \quad (3.44)$$

Adam banyak digunakan karena memberikan stabilitas dalam pelatihan jaringan saraf dalam, termasuk GNN.

3.7. Pembelajaran Semi-Terawasi (*Semi-Supervised Learning, SSL*)

Dalam praktiknya, tidak seluruh unit spasio-temporal memiliki data berlabel secara lengkap. Pada konteks empiris, hanya sebagian wilayah atau periode waktu yang memiliki nilai respon terobservasi, sedangkan sisanya tidak tersedia. Kondisi ini mendorong penerapan pembelajaran semi-terawasi (*semi-supervised learning, SSL*), yang bertujuan memanfaatkan struktur data tak berlabel guna meningkatkan proses estimasi parameter dan memperkuat generalisasi model. Pendekatan ini menjadi relevan terutama ketika data spasio-temporal memiliki keterhubungan yang kuat antarlokasi maupun antarperiode, sehingga informasi dari titik berlabel dapat disebarkan melalui struktur graf ke titik yang belum berlabel.

3.7.1. Kerangka Dasar dan Asumsi Fundamental

Secara umum, pembelajaran semi-terawasi memanfaatkan dua himpunan data, yaitu himpunan data berlabel $\mathcal{D}_\ell = \{(\mathbf{x}_i, y_i)\}_{i=1}^\ell$ dan himpunan data tak berlabel $\mathcal{D}_u = \{\mathbf{x}_j\}_{j=\ell+1}^n$, dengan $u = n - \ell \gg \ell$. Tujuan SSL adalah mempelajari fungsi prediksi $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ yang memanfaatkan informasi dari kedua himpunan tersebut. Formulasi optimisasi secara umum dituliskan sebagai

$$\min_{\theta} \left[\frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}_{\text{sup}}(f_\theta(\mathbf{x}_i), y_i) + \lambda_u \mathcal{L}_{\text{unsup}}(f_\theta; \mathcal{D}_u) + \lambda_r \mathcal{R}(\theta) \right], \quad (3.45)$$

dengan \mathcal{L}_{sup} merupakan fungsi kerugian terawasi, $\mathcal{L}_{\text{unsup}}$ merupakan fungsi kerugian yang mengekstraksi informasi dari data tak berlabel, dan $\mathcal{R}(\theta)$ adalah regularisasi parameter model. Parameter λ_u dan λ_r mengatur keseimbangan antar komponen.

3.7.2. Fungsi Kerugian Semi-Terawasi

Fungsi kerugian semi-terawasi terdiri atas dua komponen utama, yaitu komponen terawasi \mathcal{L}_{sup} dan komponen tak berlabel $\mathcal{L}_{\text{unsup}}$, yang dihubungkan melalui parameter keseimbangan λ_u . Pada regresi spasio-temporal, komponen terawasi tetap menggunakan *mean squared error* (MSE) sebagaimana pada fungsi objektif

sebelumnya:

$$\mathcal{L}_{\text{sup}} = \frac{1}{\ell} \sum_{(i,t) \in \mathcal{D}_\ell} (y_{i,t} - \hat{y}_{i,t})^2. \quad (3.46)$$

Komponen tak berlabel $\mathcal{L}_{\text{unsup}}$ dapat dirancang dalam berbagai bentuk sesuai dengan asumsi yang ingin dimanfaatkan dari data tak berlabel. Beberapa pendekatan utama yang digunakan dalam penelitian ini meliputi kerugian entropi, kerugian konsistensi, dan regularisasi berbasis graf Laplacian .

Kerugian entropi mendorong model agar menghasilkan prediksi yang lebih pasti (berentropi rendah) pada data tak berlabel. Fungsinya dituliskan sebagai

$$\mathcal{L}_{\text{entropy}} = -\frac{1}{u} \sum_{(i,t) \in \mathcal{D}_u} \sum_{k=1}^p w_{i,t}^{(k)} \log w_{i,t}^{(k)}, \quad (3.47)$$

yang secara konsep serupa dengan regularisasi entropi pada bagian sebelumnya, namun di sini diterapkan pada unit yang tidak memiliki label respon. Nilai yang rendah menunjukkan distribusi bobot yang lebih tajam, yang mengindikasikan keyakinan model terhadap struktur hubungan antarvariabel maupun antarwilayah.

Pendekatan lain menggunakan prinsip konsistensi (*consistency regularization*), yang berasumsi bahwa prediksi model seharusnya stabil terhadap gangguan kecil pada *input* atau representasi. Jika $\tau(\cdot)$ menyatakan operator augmentasi atau perturbasi (misalnya penambahan *noise* atau *dropout*), maka kerugian konsistensi dapat diformulasikan sebagai

$$\mathcal{L}_{\text{consistency}} = \frac{1}{u} \sum_{(i,t) \in \mathcal{D}_u} \left\| \hat{y}_{i,t} - \hat{y}_{i,t}^{(\tau)} \right\|^2, \quad (3.48)$$

dengan $\hat{y}_{i,t}^{(\tau)}$ adalah hasil prediksi untuk data yang telah diperturbasi. Kerugian ini mendorong model untuk menghasilkan prediksi yang tidak berubah secara signifikan terhadap variasi kecil, sehingga menghasilkan representasi yang lebih halus dan tahan terhadap gangguan.

Untuk memanfaatkan struktur graf yang mendasari data spasio-temporal, diterapkan pula regularisasi berbasis Laplacian graf. Diberikan matriks bobot $\mathbf{W} =$

$[w_{ij}]$ dan Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$, regularisasi ini dituliskan sebagai

$$\mathcal{L}_{\text{graph}} = \frac{1}{2} \sum_{i,j} w_{ij} \|\hat{y}_i - \hat{y}_j\|^2 = \hat{\mathbf{y}}^\top \mathbf{L} \hat{\mathbf{y}}, \quad (3.49)$$

yang menegakkan asumsi kehalusan pada manifold graf. Jika dua simpul terhubung dengan bobot besar, maka selisih prediksi antara keduanya didorong agar kecil. Dengan demikian, prediksi pada wilayah yang berdekatan atau periode waktu yang berurutan akan memiliki pola yang konsisten.

Berdasarkan ketiga pendekatan tersebut, komponen tak berlabel yang digunakan dalam model ini dirumuskan sebagai kombinasi linear

$$\mathcal{L}_{\text{unsup}} = \alpha_1 \mathcal{L}_{\text{entropy}} + \alpha_2 \mathcal{L}_{\text{consistency}} + \alpha_3 \mathcal{L}_{\text{graph}}, \quad (3.50)$$

dengan α_1 , α_2 , dan α_3 sebagai bobot penyeimbang yang ditentukan melalui validasi.

3.7.3. Integrasi SSL ke dalam Pelatihan GNN-GTVC dan GNN-GTWR

Integrasi SSL ke dalam model GNN dilakukan dengan menambahkan komponen kerugian tak berlabel (3.50) ke dalam fungsi objektif utama. Fungsi kerugian total untuk model semi-terawasi dapat dituliskan sebagai

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + \lambda_u \mathcal{L}_{\text{unsup}} + \lambda_r \mathcal{R}(\theta), \quad (3.51)$$

dengan $\mathcal{R}(\theta)$ mencakup regularisasi entropi dan kehalusan spasio-temporal sebagaimana telah dijelaskan pada Bagian 4.7. Dalam implementasinya, parameter λ_u mengatur sejauh mana informasi dari data tak berlabel memengaruhi pembaruan parameter jaringan.

Pada model GNN-GTVC, kerugian tak berlabel diterapkan terutama pada vektor bobot $\mathbf{w}_{i,t}$, sehingga pembelajaran pada simpul yang tidak memiliki nilai respon tetap dapat diarahkan oleh pola kedekatan antarunit yang teramati dalam graf. Sementara itu, pada model GNN-GTWR, komponen $\mathcal{L}_{\text{unsup}}$ berperan untuk

menstabilkan matriks bobot antarobservasi $\mathbf{W}_{i,t}$, agar tetap mengikuti pola spasio-temporal yang halus meskipun sebagian besar titik tidak berlabel. Dengan demikian, informasi dari simpul berlabel dapat menyebar secara halus ke simpul tak berlabel melalui proses propagasi graf.

Proses estimasi titik di luar dari data latih dilakukan dengan menggunakan pelatihan ulang pada semua data. Hal ini berarti graf dibentuk dengan melihat data latih dan data uji, tetapi variabel y dari data uji tidak digunakan untuk melatih model. Pendekatan ini digunakan juga oleh Yin dkk., 2024.

BAB IV

STUDI KASUS

4.1. Catatan Penting

ISINYA YAA PEMBAHASAN STUDI KASUS KAMUUUU.

4.2. Pembahasan

4.2.1. Pembahasan 1

Dalam membuat tabel, disarankan memakai "longtable" agar tabel nya bisa dipotong halaman. Biar lebih mudah kamu bisa pakai web https://www.tablesgenerator.com/latex_tables#google_vignette.

Contoh input gambar



Gambar 4.1 SKRIPSI TU DIKERJAIN

4.2.2. Pembahasan 2

Beberapa catatan penulisan yang wajib diperhatikan sebagai berikut.

- Penggunaan kata 'adalah' dan 'merupakan'

- Penulisan tanda baca seperti titik yang wajib ada di setiap akhir persamaan
- Dalam kalimat '... dihitung menggunakan persamaan berikut.' di akhir kata berikut harus ada 'titik'
- Kamu bisa pakai '`\begin{equation}`' atau '`$$`' atau '`$$ \begin{aligned} ... \end{aligned} $$`' dalam menuliskan persamaan, tinggal pilih mana yang kamu butuhkan. Contoh

$$\begin{aligned}
 (\mathbf{A})_{22} &= (\mathbf{C})_{22} + \min\{(\mathbf{A})_{2(2-1)}, (\mathbf{A})_{(2-1)2}, (\mathbf{A})_{(2-1)(2-1)}\} \\
 &= (\mathbf{C})_{22} + \min\{(\mathbf{A})_{21}, (\mathbf{A})_{12}, (\mathbf{A})_{11}\} \\
 &= 2 + \min\{3, 5, 2\} \\
 &= 2 + 2 = 4.
 \end{aligned}$$

$$(\mathbf{A})_{11} = (\mathbf{C})_{11}. \quad (4.1)$$

$$A = (5, 6, 5, 7, 6, 6, 6, 6) \quad B = (7, 8, 6, 10, 10, 10, 8, 8)$$

- Pemanggilan persamaan, tabel, dan gambar, WAJIB menggunakan huruf besar di depan. Contohnya adalah '... dapat dihitung menggunakan Persamaan xx', 'Dari Tabel xx', 'Berdasarkan Gambar xx'

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan pembahasan sebelumnya serta analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut.

1. Kesimpulan 1.
2. Kesimpulan 2.
3. Kesimpulan 3.

5.2. Saran

Saran untuk penelitian selanjutnya

DAFTAR PUSTAKA

- Amri, M. Z. A. (2020). Perbandingan model star dan gstar untuk peramalan indeks harga konsumen di kota padang, pekanbaru, jambi, dan palembang. *MEDI-AN*, 4(1), 30–37.
- Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic Publishers. doi: 10.1007/978-94-015-7799-1
- Axler, S. (2024). *Linear Algebra Done Right* (4th ed.). Cham: Springer. doi: 10.1007/978-3-031-41026-0
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828. doi: 10.1109/TPAMI.2013.50
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press. Diambil dari <https://www.microsoft.com/en-us/research/publication/neural-networks-for-pattern-recognition/>
- Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). *Geometric deep learning: Grids, groups, graphs, geodesics, and gauges*. Diambil dari <https://arxiv.org/abs/2104.13478>
- Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.
- Chung, F. R. K. (1997). *Spectral graph theory*. Providence, RI: American Mathematical Society.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2, 303–314. doi: 10.1007/BF02551274
- Dhrymes, P. J. (2013). *Mathematics for Econometrics* (4th ed.). New York: Springer. doi: 10.1007/978-1-4614-8145-4

- Fan, J., & Zhang, W. (2008). Statistical methods with varying coefficient models. *Statistics and Its Interface*, 1(1), 179–195. (A comprehensive review of varying coefficient models)
- Fotheringham, A. S., Brunsdon, C., & Charlton, M. (2002). . Chichester: Wiley.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press. Diambil dari <https://www.deeplearningbook.org>
- Grandvalet, Y., & Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems* (Vol. 17, pp. 529–536).
- Greene, W. H. (2002). *Econometric Analysis* (5th ed.). Upper Saddle River, New Jersey: Pearson Education.
- Greub, W. (1978). *Multilinear algebra* (2nd ed.). New York: Springer-Verlag.
- Gujarati, D. N., & Porter, D. C. (2009). *Basic Econometrics*. New York: McGraw-Hill/Irwin.
- Hackbusch, W. (2012). *Tensor Spaces and Numerical Tensor Calculus*. Berlin Heidelberg: Springer. doi: 10.1007/978-3-642-28027-6
- Hagenauer, J., & Helbich, M. (2022). Geographically weighted artificial neural networks: A framework for non-linear local modelling. *International Journal of Geographical Information Science*, 36(2), 231–252.
- Hamilton, W., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *31st conference on neural information processing systems (neurips)*. Diambil dari <https://arxiv.org/abs/1706.02216>
- Hastie, T., & Tibshirani, R. (1993). Varying-coefficient models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(4), 757–796.
- Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Upper Saddle River, NJ: Pearson Education / Prentice Hall.
- Horn, R. A., & Johnson, C. R. (2012). *Matrix Analysis* (2nd ed.). Cambridge: Cambridge University Press. doi: 10.1017/CBO9781139020411
- Huang, B., Wu, B., & Barry, M. (2010). Geographically and temporally weighted

- regression for modeling spatio-temporal variation in house prices. *International Journal of Geographical Information Science*, 24(3), 383–401.
- Jiang, T.-X., Huang, T.-Z., Zhao, X.-L., & Deng, L.-J. (2020, July). Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. *Journal of Computational and Applied Mathematics*, 372, 112680. Diambil dari <http://dx.doi.org/10.1016/j.cam.2019.112680> doi: 10.1016/j.cam.2019.112680
- Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th international conference on learning representations (iclr)*. Diambil dari <https://arxiv.org/abs/1609.02907>
- Lang, S. (1987). *Linear Algebra* (3rd ed.). New York: Springer. doi: 10.1007/978-1-4757-1949-9
- LeSage, J. P., & Pace, R. K. (2009). *Introduction to Spatial Econometrics*. Boca Raton, FL: CRC Press.
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). (5th ed.). Hoboken, NJ: Wiley.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (icml)* (pp. 807–814).
- Ni, X., Wu, H., & Zhang, X. (2022). Sannwr: Spatial and attribute neural network weighted regression for modeling complex spatial heterogeneity. *International Journal of Geographical Information Science*, 36(7), 1401–1426.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *6th international conference on learning representations (iclr)*. Diambil dari <https://arxiv.org/abs/1710.10903>
- Xiong, J., Zhang, H., Wang, X., & Liu, Y. (2024). Generalized spatial-temporal regression graph convolutional transformer: A hybrid framework for spatio-temporal prediction. *Applied Intelligence*. doi: 10.1007/s40747-024-01578

- Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? In *7th international conference on learning representations (iclr)*. Diambil dari <https://arxiv.org/abs/1810.00826>
- Yin, P., Huang, B., Du, Z., Wu, H., et al. (2024). Gnnwr: An r package for geographically neural network weighted regression and its spatiotemporal extension. *Journal of Geographical Systems*.
- Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence (ijcai-18)* (pp. 3634–3640).
- Zhu, D., Liu, Y., Yao, X., & Fischer, M. M. (2022). Spatial regression graph convolutional neural networks (srcnns). *GeoInformatica*, 26(3), 557–582.

LAMPIRAN A

Data

LAMPIRAN B

Syntax R

1 SYNTAX R ALALALALALLALA