Instituto de MATEMÁTICA E ESTATÍSTICA UFRGS

DEPARTAMENTO DE ESTATÍSTICA
E-mail: dest@mat.ufrgs.br

Trabalho de Conclusão de Curso

Modelagem Geoestatística de Extremos de Precipitação: Comparação entre o Estimador de Máxima Verossimilhança e Redes Neurais de Grafo

Yuri Dessimon

19 de dezembro de 2025

Yuri Dessimon

Modelagem Geoestatística de Extremos de Precipitação: Comparação entre o Estimador de Máxima Verossimilhança e Redes Neurais de Grafo

Trabalho de Conclusão apresentado à comissão de Graduação do Departamento de Estatística da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Bacharel em Estatística.

Orientadora: Prof. Dr. Márcia Helena Barbian

Porto Alegre
12/12/2025

Yuri Dessimon

Modelagem Geoestatística de Extremos de Precipitação: Comparação entre o Estimador de Máxima Verossimilhança e Redes Neurais de Grafo

Este Trabalho foi julgado adequado para obtenção dos créditos da disciplina Trabalho de Conclusão de Curso em Estatística e aprovado em sua forma final pelo Orientadora e pela Banca Examinadora.

Orientadora:_____
Prof. Dr. Márcia Helena Barbian, UFMG
Doutora pela Universidade Federal de Minas Gerais, Belo Horizonte, MG

Banca Examinadora:

Dr. João Henrique Ferreira Flores, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, RS

Dr. Cristiano Lima Hackmann, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, RS

Porto Alegre
12/12/2025

"It's not down on any maps;
true places never are"

Herman Melville, Moby-Dick or, The Whale

Agradecimentos

Agradeco à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) pelo apoio financeiro, por meio do Projeto nº 24/2551-0002361-5, contemplado no Edital 06/2024 – Programa de Pesquisa e Desenvolvimento Voltado a Desastres Climáticos. Este suporte foi fundamental para a realização e o avanço desta pesquisa.

Agradeço a minha orientadora Márcia por me ajudar nesse trabalho e por todo o conhecimento transmitido durante a minha graduação

Aos professores Cristiano e João que receberam esse trabalho de bom agrado e pelas ótimas sugestões.

Muito obrigado a todos que fizeram alguma parte da minha vida durante a graduação, eu valorizo vocês mais do que imaginam.

Muito obrigado ao Calvin (meu gato), eu amo ele.

Resumo

A predição de variáveis climáticas enfrenta desafios devido à complexidade e não-estacionariedade dos fenômenos atmosféricos, limitações que frequentemente desafiam a geoestatística clássica. Este trabalho investiga o desempenho da NN-GLS, uma arquitetura híbrida que une a flexibilidade das redes neurais para modelar tendências não-lineares à estrutura de correlação espacial. O objetivo principal foi avaliar sua capacidade preditiva em comparação ao método do estimador de máxima verossimilhança (EMV). A metodologia utilizou dados meteorológicos reais e simulações de Monte Carlo. Os resultados indicaram que, em cenários estacionários e simples, o método EMV permanece superior e mais estável. A NN-GLS apresentou tendência de subestimar o erro aleatório dos dados. No entanto, a rede apresentou melhora nos resultados conforme a não linearidade das simulações aumenta.

Palavras-Chave: Geoestatística, Processo Gaussiano, Aprendizado de Máquina, Redes Neurais de Grafos, Dados Climáticos.

Abstract

The prediction of climatic variables faces challenges due to the complexity and non-stationarity of atmospheric phenomena, limitations that often defy classical geostatistics. This work investigates the performance of NN-GLS, a hybrid architecture that combines the flexibility of neural networks for modeling non-linear trends with spatial correlation structures. The main objective was to evaluate its predictive capacity compared to the Maximum Likelihood Estimator (MLE) method. The methodology employed real meteorological data and Monte Carlo simulations. The results indicated that, in simple and stationary scenarios, the MLE method remains superior and more stable. The NN-GLS showed a tendency to underestimate the random error of the data. However, the network demonstrated improved results as the non-linearity of the simulations increased.

Keywords: Geostatistics, Gaussian Processes, Machine Learning, Graph Neural Networks, Climate Data.

Sumário

1 Introdução **12**

2 O Campo Gaussiano e a Modelagem Geoestatística **14**
2.1 Processo Geoestatístico . . . 14
2.1.1 Processos Gaussiano . . . 14
2.1.2 O Modelo . . . 15
2.1.3 Estacionariedade . . . 16
2.2 Semivariograma . . . 17
2.2.1 Semivariograma Amostral . . . 17
2.3 O Semivariograma . . . 17
2.4 Modelos de Variograma e Função de Covariância . . . 18
2.4.1 Esférico . . . 18
2.4.2 Exponencial . . . 19
2.4.3 Matérn . . . 19
2.4.4 Estimador de Máxima Verossimilhança . . . 19
2.5 Krigagem . . . 20
2.5.1 Krigagem Simples e Ordinária . . . 21
2.5.2 Krigagem Universal . . . 21
2.5.3 Estimação de Parâmetros na Krigagem Universal . . . 22

3 GNN **23**
3.1 Grafo . . . 23
3.1.1 Representação Matricial . . . 23
3.2 Graph Embedding . . . 25
3.3 Níveis de Predição . . . 25
3.4 Camada de uma GNN . . . 26
3.5 Arquitetura . . . 27
3.6 Treinamento . . . 27
3.7 Desafios da Estimação na GNN . . . 28

4 NN-GLS **29**
4.1 Fundamentos Iniciais . . . 29
4.1.1 OLS x GLS . . . 29
4.2 Modelo teórico . . . 29
4.3 NN-GLS como uma GNN . . . 30
4.4 Arquitetura . . . 31
4.5 Treinamento . . . 31

4.6 Krigagem 32

5 Simulação **33**
5.1 Pré-Processamento dos Dados 33
5.2 Estudo de Simulações 33
5.2.1 Cenário 1 33
5.2.2 Comparação dos Resultados - Cenário 1 34
5.2.3 Cenário 2 39
5.2.4 Comparação dos Resultados - Cenário 2 39
5.2.5 Estimação dos Parâmetros 41
5.2.6 Desempenho (EQM) 41

6 Dados Reais **43**
6.1 Arquitetura da rede 43
6.2 Resultados 44

7 Conclusão **47**

Referências **47**

Lista de Figuras

2.1 Imagem de um semivariograma. 18
2.2 Gráfico das 3 funções de correlação. 20

3.1 Pintura das 7 pontes de Königsberg. 24
3.2 Grafo das sete pontes. 24
3.3 diferente tipos de embeddings de um grafo. 26
3.4 Arquitetura de uma GNN. 28

4.1 Arquitetura da NN-GLS. 32

5.1 Exemplos de diferentes campos gaussianos simulados com parâmetros para a função de covariância $\sigma^2 = 3$ e $\phi = 5$, mas diferentes estruturas de média: Constante, Linear e Complexa. 34
5.2 Gráficos de comparação dos valores estimados dos parâmetros e do EQM. 42

6.1 Mapa da predição da quantidade total de precipitação no período de 01/05/2024 até 01/05/2025. na esquerda EMV e na direita NN-GLS. 44
6.2 Gráfico do EQM da krigagem na esquerda EMV e na direita NN-GLS. 44
6.3 Gráficos de dependências parciais da NN-GLS em ordem, de esquerda para direita, das covariáveis: pressão, temperatura e umidade. 46

Lista de Tabelas

5.1 Cenário 1: estimativas dos parâmetros considerando $\sigma^2 = 1$, $\tau^2 = 1$ e $phi = (1, 5, 8, 10)$. 34
5.2 Resumo Comparativo de Estimação - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 3$). 35
5.3 Resumo Comparativo de Estimação - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 5$). 35
5.4 Resumo Comparativo de Estimação - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 10$). 36
5.5 Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 1$). 37
5.6 Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 3$). 37
5.7 Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 5$). 38
5.8 Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 10$). 38
5.9 Resumo Comparativo de Estimação (EMV) - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 1$) 39
5.10 Resumo Comparativo de Estimação (EMV) - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 5$) 40
5.11 Comparação de EQM - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 1$) 40
5.12 Comparação de EQM - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 5$) 40

1 Introdução

Condições climáticas extremas são fenômenos raros que ocorrem em lugares e épocas do ano específicos, tais como ondas de calor e frio, secas e inundações, grandes tempestades. A National Academies of Sciences and Medicine (2016) define tais eventos como aqueles que possuem uma probabilidade extremamente baixa de serem observados dado um determinado período de tempo. Entre as as maiores causas de mudanças climáticas no mundo estão a urbanização, a agricultura, a poluição de processos industriais e o aumento de dióxido de carbono (CO2) Kabir et al. (2023)).

Segundo Wang et al. (2022), as mudanças climáticas causam o aumento da temperatura da superfície da terra todo o ano. Devido a essas alterações, a ocorrência de condições climáticas extremas tem se tornado cada vez mais frequente no mundo. Como esses eventos são inerentemente fenômenos espaciais, em que a observação em um local é correlacionada com a de locais próximos, é interessante levar em conta essa dependência espacial.

Dados espaciais são frequentemente modelados como realizações de um processo estocástico (Cressie, 2015). Embora a geoestatística tenha se consolidado através do estudo de propriedades do solo, seu uso se expande à outras áreas, sendo aplicada a dados climáticos. Um exemplo é o trabalho de Lanfredi et al. (2015), que estima a temperatura na Europa. Esses métodos muitas vezes assumem a estacionariedade dos dados, que é um pressuposto que pode limitar a capacidade de capturar dinâmicas não lineares complexas, frequentemente observadas em eventos climáticos extremos (Maraun et al., 2010).

Como alternativa para modelar estruturas complexas de covariância em dados ambientais, surgem as redes neurais artificiais. Por exemplo, no artigo Knutti et al. (2003) os autores aplicam uma rede neural probabilística para predizer o aquecimento da superfície da água do oceano, já os autores Ashrafi et al. (2012) utilizam uma rede neural perceptron multicamadas (MLP), para predizer o aumento de temperatura devido à mudanças climáticas no Irã. O aumento do uso dessas técnicas se dá pela facilidade que esses métodos têm de ajustar modelos complexos não lineares. Apesar dessa flexibilidade, esses modelos falham ao tratar as entradas como independentes, ignorando a estrutura de dependência espacial fundamental para dados climatológicos.

Os exemplos de redes neurais artificiais anteriores não levam em conta a dependência entre as observações; os trabalhos a seguir tentam considerar mais à fundo essas relações. Como Jiang et al. (2022) que aplicaram Redes Neurais Convolucionais Profundas (DCNN) para prever o momento e a localização de eventos extremos em sistemas físicos bidimensionais. A abordagem, focada no aprendizado direto a

partir dos dados sem depender de equações físicas explícitas, foi validada com imagens sintéticas e medições reais de velocidade do vento no Atlântico Norte. Nessa mesma linha, visando capturar dependências espaço-temporais simultâneas, Kim et al. (2019) utilizaram a arquitetura ConvLSTM para prever e rastrear a trajetória de furacões, desenvolvendo modelos específicos que superaram o desempenho de métodos convencionais baseados em regiões (R-CNNs).

Em 2009, Scarselli et al. (2009) propuseram o modelo de Redes Neurais em Grafos (GNN), um método que incorpora a relação das variáveis pela estrutura do grafo. Essa proposta básica da GNN possibilitou a análise de dependências complexas e não lineares entre as observações, permitindo indicar a estrutura espacial dentro de uma rede neural através das conexões do grafo. Essa capacidade de representar relacionamentos foi a base para a construção de outras arquiteturas, como as Redes Convolucionais em Grafos (GCN) (Sanchez-Lengeling et al., 2021b) e as Redes de Atenção em Grafos (GAT) (Velickovic et al., 2017).

Tendo em vista o problema e as dificuldades de predizer condições climáticas extremas, esse trabalho busca investigar como o uso de um modelo híbrido, que combina a complexidade de redes neurais com a geoestatística pode ajudar a prever esses eventos. Situando-se na interface entre o aprendizado profundo e a modelagem espacial tradicional, o presente trabalho visa comparar a arquitetura Neural Network with Generalized Least Squares (NN-GLS), proposta por Zhan and Datta (2025) que integra Redes Neurais de Grafos (GNN) ao formalismo geoestatístico, com a modelagem baseada em Processos Gaussianos clássicos. Considerando os desafios inerentes à predição de condições climáticas extremas, o objetivo principal é investigar a capacidade da NN-GLS em dados reais e simulados, avaliando seu desempenho preditivo comparativamente ao método geoestatístico tradicional utilizando o estimador de máxima verossimilhança (EMV) (Diggle et al., 2007). A hipótese central é que a NN-GLS, por sua aptidão em modelar dinâmicas não lineares complexas sem descartar a estrutura de dependência espacial, forneça estimativas mais precisas.

A estrutura deste trabalho começa com uma revisão dos conceitos de geoestatística, apresentando as bases teóricas de dependência espacial e processos gaussianos necessárias para a modelagem. Na sequência, são apresentadas as GNNs, fornecendo a base necessária para compreender a arquitetura híbrida utilizada. Posteriormente, o texto apresenta a formulação da NN-GLS, integrando os dois campos de estudo anteriores. O trabalho conclui com a análise de simulação, em que se avalia a eficácia da abordagem proposta frente ao EMV e termina com a aplicação a um conjunto de dados reais.

2 O Campo Gaussiano e a Modelagem Geoestatística

A análise espacial baseia-se no princípio de que observações próximas no espaço tendem a apresentar valores mais semelhantes entre si. Este conceito foi pioneiramente desenvolvido nos trabalhos empíricos de Daniel G. Krige (Krige, 1951), voltados à estimativa de reservas minerais, e posteriormente formalizado por Georges Matheron (Matheron, 1963). Assim, a geoestatística consolidou-se como uma abordagem quantitativa para modelar e prever processos aleatórios contínuos.

Este capítulo apresenta a definição de Processos Gaussianos (Campos Aleatórios Gaussianos), estrutura probabilística para caracterizar a dependência entre observações, e técnicas de inferência e predição, com ênfase na função de covariância, no estimador de máxima verossimilhança e na krigagem.

A modelagem de fenômenos que variam continuamente no espaço fundamenta-se na teoria de processos estocásticos. Neste contexto, assume-se que os dados observados constituem apenas uma das possíveis realizações desse processo aleatório subjacente. O processo gaussiano, que desempenha um papel central na inferência espacial devido às suas propriedades analíticas tratáveis, é apresentado a seguir.

2.1 Processo Geoestatístico

2.1.1 Processos Gaussiano

Considere um processo estocástico espacialmente contínuo e de valores reais $\{\mathbf{Z}(s) : s \in \mathbb{R}^2\}$, no qual $\mathbf{Z}(s)$ representa o valor da variável de interesse na localização s. Assume-se que esse processo seja diferenciável, ou mesmo muitas vezes diferenciável, conforme as necessidades do modelo. Dentro desse contexto, o modelo mais simples e amplamente utilizado que satisfaz tais hipóteses é o processo gaussiano.

Um processo estocástico $\mathbf{Z}(s)$ é um campo Gaussiano se a distribuição conjunta de $Z(s_1), \ldots, Z(s_n)$ tiver distribuição normal multivariada, $\mathbf{Z}(s) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, como na equação 2.1:

$$f(z(s_1), \ldots, z(s_n)) = \frac{\exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z} - \boldsymbol{\mu})\right)}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}}, \tag{2.1}$$

para qualquer n conjunto de localizações s_i (Diggle et al. (2003)). O processo é estacionário se a esperança de $\mathbf{Z}(s)$ for a mesma para todo s, a variância de $\mathbf{Z}(s)$

for a mesma para todo s e a correlação entre $Z(s_i)$ e $Z(s_j)$ depender apenas de $h = ||s_i - s_j||$, a distância Euclidiana entre s_i e s_j. A definição de estacionariedade será vista com mais profundidade nas sessões a seguir.

2.1.2 O Modelo

Para fins de análise e predição, a metodologia geoestatística busca descrever a estrutura de variabilidade espacial contida no campo aleatório $\mathbf{Z}(s)$ (Equação 2.1). A abordagem clássica para modelar este campo é através da sua decomposição em componentes que representam diferentes escalas de variação (Cressie, 2015). Esta decomposição é essencial para isolar o componente de dependência espacial de outros fatores, como uma tendência de larga escala ou a variabilidade puramente aleatória.

O modelo geoestatístico assume que a realização observada do campo aleatório $Z(s)$ em qualquer localização s pode ser decomposta em três componentes aditivos. Essa decomposição pode ser expressa pela equação 2.2 (Moraga, 2019):

$$Z(s) = \mu(s) + \omega(s) + \epsilon(s), \tag{2.2}$$

- $\mu(s)$ representa a tendência. Este é o componente determinístico do modelo, definido como o valor esperado do processo, tal que $\mu(s) = E[Z(s)]$. Ele é responsável por capturar variações de grande escala e padrões estruturais ou globais nos dados, como um gradiente climático regional ou uma alteração geológica sistemática.

- $\omega(s)$ é um processo estocástico de pequena escala. Assume-se que este processo seja estacionário com média zero ($E[\omega(s)] = 0$) e que capture a estrutura de dependência espacial (autocorrelação) nos dados. A dependência entre dois pontos quaisquer é modelada por uma matriz de covariância espacial $\mathbf{\Sigma}$, cujos elementos Σ_{ij} são definidos em função da distância euclidiana $h_{ij} = ||s_i - s_j||$ entre as localizações e uma função de correlação $C(h; \phi)$:

  $$\mathbf{\Sigma}_{ij} = \text{Cov}(\omega(s_i), \omega(s_j)),$$

  em que $C(\cdot)$ é uma função de correlação positiva definida que decai conforme a distância aumenta, controlada pelo parâmetro de alcance ϕ. A função de correlação e o parâmetro ϕ serão explicados nas Seções 2.3 e 2.4, respectivamente.

- $\epsilon(s)$ é um processo não correlacionado espacialmente, também com média zero ($E[\epsilon(s)] = 0$) e variância constante ($Var[\epsilon(s)] = \tau^2$). Este termo representa o erro de medição ou a variabilidade em uma micro-escala conhecida como efeito pepita.

O EMV, como será visto na Seção 2.4.4, modela a tendência $\mu(s)$ como uma função linear de covariáveis. Enquanto, o modelo NN-GLS, propõe modelar $\mu(s)$ usando uma função não-linear. Ambos os métodos, no entanto, utilizam a estrutura de variância e efeitos aleatórios da Equação 2.1 para realizar predições.

2.1.3 Estacionariedade

Como mostrado na definição de processo gaussiano, em geoestatística é importannte a definição de estacionariedade. A forma mais rigorosa é a estacionariedade estrita. Um campo aleatório $Z(s)$ é dito estritamente estacionário se a distribuição de probabilidade conjunta de qualquer conjunto de n variáveis $\{Z(s_1), \ldots, Z(s_n)\}$ for invariante sob translação espacial. Ou seja, para qualquer vetor de deslocamento h e qualquer número inteiro n, temos:

$$F(Z(s_1), \ldots, Z(s_n)) = F(Z(s_1 + h), \ldots, Z(s_n + h)). \tag{2.3}$$

O processo gaussiano é estritamente estacionário.

A suposição mais comum é a estacionariedade de segunda ordem, que postula que as propriedades estatísticas de primeira e segunda ordem do campo aleatório $Z(s)$ dependem apenas da distância h.

Cressie (2015) define estacionariedade de segunda ordem para um campo aleatório $\mathbf{Z}(s)$ como:

i. A média do processo é constante em qualquer localização s;

$$E[\mathbf{Z}(\mathbf{s})] = \mu;$$

ii. A covariância de 2 pontos depende apenas de sua distância para todo deslocamento h.

$$\text{Cov}[Z(s_i), Z(s_j)] = \text{Cov}[Z(s_i + h), Z(s_j + h)] =: C(h),$$

Quando $h = 0$, obtemos a variância, $C(0) = Var[Z(s)]$. Se a função $C(\mathbf{h})$ depender apenas da distância $||\mathbf{h}||$ e não da direção do vetor $\mathbf{h}$, o processo é dito isotrópico.

Essa suposição também implica a estacionariedade intrínseca, que afirma:

i. A média das diferenças entre dois pontos quaisquer s_i e $s_i + h$ é zero

$$E[Z(s_i + h) - Z(s_i)] = E[\mu - \mu] = 0;$$

ii. A variância das diferenças depende apenas do vetor de separação h:

$$Var[Z(s_i + h) - Z(s_i)] = E[Z(s_i + h) - Z(s_i)]^2 =: 2\gamma(h).$$

A função $2\gamma(h)$ é chamada de variograma e $\gamma(h)$ é o semivariograma, que mede a dissimilaridade média esperada entre observações separadas por uma distância h. A estacionariedade de segunda ordem implica em estacionariedade intrínseca, como é demonstrado abaixo:

$$2\gamma(h) = Var[Z(s+h) - Z(s)] = Var[Z(s+h)] + Var[Z(s)] - 2 \cdot Cov[Z(s+h), Z(s)],$$

$$2\gamma(h) = C(0) + C(0) - 2C(h) = 2(C(0) - C(h)),$$

que nos leva a $\gamma(h) = C(0) - C(h)$. Esta equação demonstra que, para um processo que satisfaz a estacionariedade de segunda ordem, o semivariograma $\gamma(h)$ é simplesmente uma função obtida das funções de covariâncias.

2.2 Semivariograma

2.2.1 Semivariograma Amostral

Na seção anterior, o semivariograma $\gamma(h)$ foi definido como uma função teórica. Na prática, esta função é desconhecida e deve ser estimada a partir dos n dados observados $\{Z(s_i), ..., Z(s_n)\}$. O resultado dessa estimação é o semivariograma amostral (ou empírico). Sob a Hipótese Intrínseca, o estimador de momentos do semivariograma (Cressie (2015)) é dado por:

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{(i,j)\in N(h)} [Z(s_i) - Z(s_j)]^2,$$

na qual $N(h) = (s_i, s_j) : s_i - s_j = h$ é o conjunto de todos os vizinhos separados pela distância h.

Esse estimador tem 2 problemas essenciais Cressie (2015): o primeiro é que ele não satisfaz a propriedade de que a matriz de correlação gerada deve ser positiva definida, e o segundo é que ele só estima a correlação em distâncias observadas. $\hat{\gamma}(h)$ fornece valores de semivariância apenas para um conjunto discreto de classes de distância h. A predição em geoestatística, conhecida como Krigagem (Seção 2.5), requer valores de covariância para as distâncias exatas entre cada par de pontos de dados e entre os dados e o ponto de predição s_0. A solução para ambos os problemas é ajustar um modelo teórico paramétrico $\gamma(h;\theta)$ para $\hat{\gamma}$, garantindo que valores de covariância para todas distâncias e função de covariância $C(h) = C(0) - \gamma(h)$ correspondente seja positiva definida.

2.3 O Semivariograma

Podemos definir o variograma pelas funções de covariância de forma que ele dependa de 3 principais parâmetros. O parâmetro patamar (Sill, $C(0)$), o alcance (range, ϕ) e o efeito pepita ($\epsilon(s)$). Bohling (2005) introduziu esses parâmetros como:

- τ^2 representa a variabilidade do efeito pepita $\epsilon(s)$ que não pode ser explicada pela estrutura espacial ($\omega(s)$). Esse parâmetro aparece no variograma (ou covariância) como uma aparente descontinuidade na origem, $\lim_{h\to 0} \gamma(h) = \tau^2$.
- O parâmetro σ^2, também é conhecido como variância do processo, é o valor em que a semivariância permanecerá constante depois de certa distância h, também conhecido como patamar parcial (partial sill).
- O patamar é composto pelo efeito pepita e pelo patamar parcial (σ^2), de modo que $\lim_{h\to \text{inf}} \gamma(h) \to \lim_{h\to 0} C(h) = \tau^2 + \sigma^2$.
- O alcance ϕ (range), indica a distância na qual dois pontos podem ser considerados independentes, ou seja quando o semivariograma atinge o valor patamar (sill). Assumimos que a autocorrelação é zero além dessa distância.

Podemos ver na Figura 2.1 uma representação de um variograma com patamar igual a 1, alcance igual a 4000 e efeito pepita igual a 0,2.

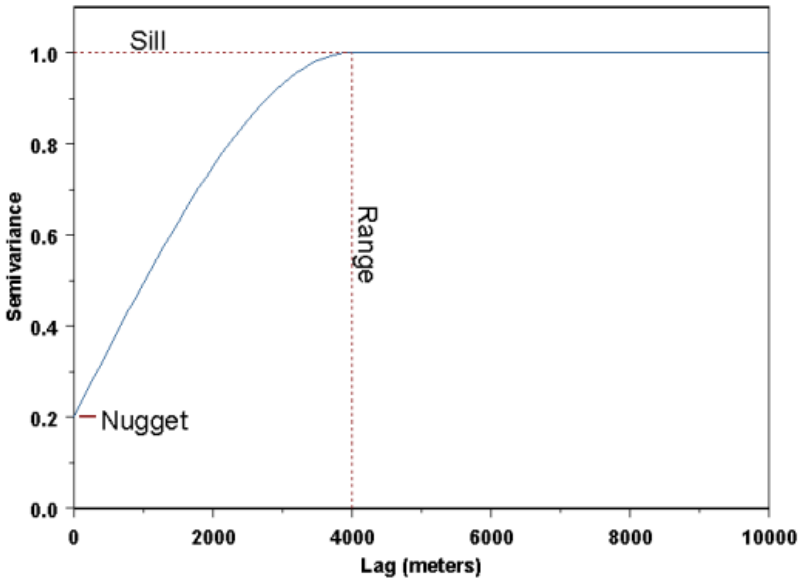

Figura 2.1: Imagem de um semivariograma.
Fonte: Bohling (2005)

O processo de modelagem, portanto, consiste em calcular $\hat{\gamma}(h)$ e, em seguida, ajustar um modelo teórico válido para estimar os parâmetros $\theta = (\phi, \sigma^2, \tau^2)$ que melhor descrevem a dependência espacial dos dados.

2.4 Modelos de Variograma e Função de Covariância

A seguir, são apresentados as funções de covariância mais comuns.

2.4.1 Esférico

O modelo Esférico é amplamente utilizado na geoestatística tradicional, em parte por possuir um alcance (ϕ) finito. Sua função de covariância $C(h)$ e semivariograma $\gamma(h)$ são definidos por Cressie (2015):

Função de Covariância:

$$C(h) = \begin{cases} \sigma^2\left(1 - \frac{3}{2}\left(\frac{|h|}{\phi}\right) + \frac{1}{2}\left(\frac{|h|}{\phi}\right)^3\right) & \text{se } 0 \leq |h| \leq \phi, \\ 0 & \text{se } |h| > \phi. \end{cases}$$

Função de Semivariograma:

$$\gamma(h;\theta) = \begin{cases} \tau^2 & \text{se } h = 0, \\ \tau^2 + \sigma^2\left(\frac{3}{2}\left(\frac{|h|}{\phi}\right) - \frac{1}{2}\left(\frac{|h|}{\phi}\right)^3\right) & \text{se } 0 < |h| \leq \phi, \\ \tau^2 + \sigma^2 & \text{se } |h| > \phi. \end{cases}$$

2.4.2 Exponencial

O modelo da função de covariância exponencial decai exponencialmente com o aumento da distância.

Função de Covariância:

$$C(h) = \sigma^2 \exp\left(-\frac{|h|}{\phi}\right).$$

Função de Semivariograma:

$$\gamma(h) = \begin{cases} \tau^2 & \text{se } h = 0, \\ \tau^2 + \sigma^2\left(1 - \exp\left(-\frac{|h|}{\phi}\right)\right) & \text{se } h > 0. \end{cases}$$

2.4.3 Matérn

Sendo K a função de Bessel modificada de segunda ordem, ν determina o grau de decaimento da correlação conforme a distância e κ é um parâmetro de escala. O valor inteiro de ν determina a diferenciabilidade quadrática média do processo subjacente, o que é importante para as predições feitas usando tal modelo. No entanto, ν é geralmente fixado, segundo Sherman (2011) é:

Função de Covariância:

$$C_\nu(h) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{h}{\rho}\right)^\nu K_\nu\left(\frac{h}{\rho}\right).$$

Função de Semivariograma:

$$\gamma(h) = \begin{cases} \tau^2 & \text{se } h = 0, \\ \tau^2 + \sigma^2\left(1 - \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{h}{\phi}\right)^\nu K_\nu\left(\frac{h}{\phi}\right)\right) & \text{se } h > 0. \end{cases}$$

Podemos ver na Figura 2.2 uma ilustração das 3 funções de correlação de matern com $\phi = 3$, $\sigma^2 = 1$, $\tau^2 = 0$ e o parâmetro de suavidade da matérn igual a $\nu = 1.5$

2.4.4 Estimador de Máxima Verossimilhança

A inferência sobre os parâmetros que definem a estrutura de dependência espacial, agrupados como $\theta = (\sigma^2, \phi, \tau^2)$, é frequentemente realizada através do método da máxima verossimilhança Diggle et al. (2007).

Na prática, maximiza-se o logaritmo da função de verossimilhança, definido por:

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}(\boldsymbol{\theta})| - \frac{1}{2}(\mathbf{Z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}(\mathbf{Z} - \boldsymbol{\mu}), \tag{2.4}$$

em que n é o número de observações, $\boldsymbol{\mu}$ é o vetor de médias e $\boldsymbol{\Sigma}$ é a matriz de covariância. O objetivo é encontrar o conjunto $\hat{\theta}$ que maximiza $\mathcal{L}(\boldsymbol{\theta})$.

Entretanto, a aplicação direta deste estimador enfrenta barreiras práticas quando o número de observações n é elevado Rasmussen and Williams (2006). Isso ocorre devido à complexidade de $O(n^3)$ computações exigida no cálculo da verossimilhança. Essa complexidade cúbica advém das operações de álgebra linear necessárias a cada iteração do algoritmo de otimização: o cálculo do determinante $|\boldsymbol{\Sigma}|$ e, principalmente, a inversão da matriz de covariância $\boldsymbol{\Sigma}^{-1}$, tornando o método proibitivo para grandes conjuntos de dados.

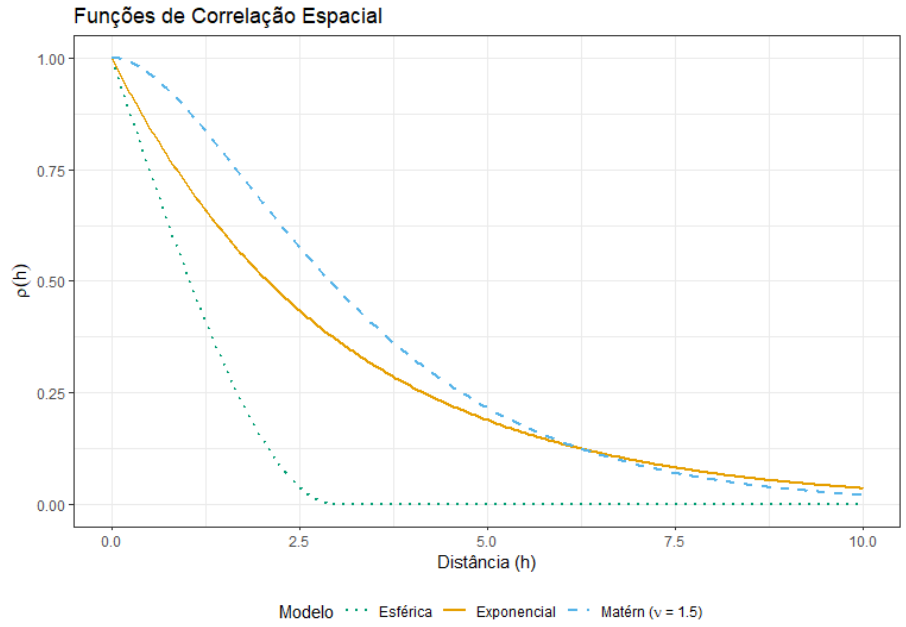

Figura 2.2: Gráfico das 3 funções de correlação.

2.5 Krigagem

O método de predição da geoestatística é conhecido como Krigagem, tal método é usado para estimar a distribuição de um campo aleatório com base em um conjunto de pontos amostrados. O método se distingue de interpoladores determinísticos, como o Inverso da Distância Ponderada, por ser um estimador estocástico que fornece não apenas a predição, mas também a variância do erro de estimação Cressie (2015).

A Krigagem é conhecida como o Melhor Preditor Linear Não-Viesado (BLUP - Best Linear Unbiased Predictor). Segundo Goovaerts (1997), essas características são definidas da seguinte forma:

- Melhor (Best): O estimador é projetado para minimizar a variância do erro de estimação (tornando-o o mais preciso possível dentro da classe de estimadores lineares);
- Linear: A predição em um local não observado é obtida através de uma combinação linear ponderada dos dados amostrais observados;
- Não-Viesado (Unbiased): Garante que a esperança matemática do erro de estimação seja zero (ou seja, em média, o estimador não subestima nem superestima o valor real).

A formulação do preditor linear busca encontrar os pesos ótimos (λ_i), que tem a propriedade $\sum \lambda_i(s) = 1$, para qualquer localização alvo s, das n amostras $Z(s_i)$, utilizadas na estimação do valor desconhecido $Z(s_0)$. Esse preditor é definido pela equação abaixo:

$$\hat{Z}(s_0) = \sum_{i=1}^{n} \lambda_i Z(s_i), \tag{2.5}$$

Para processos gaussianos, o valor da variável aleatória em um novo ponto $Z(s_0)$, conjuntamente com o vetor de observações $\mathbf{Z} = (Z(s_1), \ldots, Z(s_n))^\top$, segue uma distribuição normal multivariada (Marinescu, 2024):

$$\begin{pmatrix} Z(s_0) \\ \mathbf{Z} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_0 \\ \boldsymbol{\mu} \end{pmatrix}, \begin{pmatrix} \Sigma_{00} & \boldsymbol{\Sigma}_{01} \\ \boldsymbol{\Sigma}_{10} & \boldsymbol{\Sigma}_{11} \end{pmatrix} \right),$$

onde $\mathbf{Z}$ é um vetor de observações de dimensão n, μ_0 representa a média do processo na localização s_0, e $\boldsymbol{\mu}$ é o vetor de médias, de dimensão $n \times 1$, associado às localizações observadas. A matriz $\boldsymbol{\Sigma}_{11}$ é a matriz de covariâncias, de dimensão $n \times n$, correspondente ao vetor $\mathbf{Z}$, a qual passaremos a denotar simplesmente por $\boldsymbol{\Sigma}$.

Nessas condições, o estimador de krigagem para o valor não observado no ponto s_0, denotado por $\hat{Z}(s_0)$, é dado pela expressão:

$$\hat{\mathbf{Z}}(s_0) = E[\mathbf{Z}(\mathbf{s_0})|\mathbf{Z}(\mathbf{s})].$$

As subseções a seguir detalharão as fórmulas específicas para alguns tipo de krigagem, mostrando como a abordagem usando processo gaussiano deriva esses preditores dependendo se a média μ é assumida como conhecida, desconhecida e constante ou desconhecida e não-constante.

2.5.1 Krigagem Simples e Ordinária

Na terminologia geoestatística tradicional, assume-se que o processo estocástico $Z(s)$ pode ser decomposto em uma tendência determinística como na Equação **??**.

Quando a média $\mu(s)$ é assumida conhecida e constante, o método de predição é denominado krigagem simples (Diggle et al., 1998). Para o uso prático, contudo, frequentemente precisamos estimar os parâmetros do modelo. Uma abordagem comum é a krigagem ordinária, que se distingue da simples por tratar a média μ como uma constante desconhecida (embora os parâmetros de covariância ainda sejam assumidos conhecidos ou estimados à parte). Isso resulta em um preditor linear semelhante, com a diferença de que o valor desconhecido de μ é substituído pelo seu estimador de Mínimos Quadrados Generalizados (GLS).

$$\hat{\mu} = (\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1})^{-1}\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{Y}. \tag{2.6}$$

2.5.2 Krigagem Universal

Em alguns problemas, identificar um modelo para $\mu(s) = E[Z(s)|x(s)]$ é de interesse científico direto. Tipicamente, isso surge quando registramos um número de variáveis explicativas espaciais, $x_p(s)$, e desejamos saber quais dessas influenciam a resposta média, $\mu(s)$, ou para melhorar a predição de uma superfície (Diggle et al., 1998). Ambos os tipos de problema são abrangidos pelo modelo Gaussiano com uma especificação linear para a tendência,

$$\mu(s) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i(s) = x(s)'\beta,$$

em que $x(\mathbf{s}) = (1, x_1(s), ..., x_p(s))$. A inferência sobre os parâmetros de regressão $\boldsymbol{\beta}$ pode ser realizada com o estimador de máxima verossimilhança.

2.5.3 Estimação de Parâmetros na Krigagem Universal

Sob o modelo Gaussiano $\mathbf{Z}(s) \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{\Sigma} + \tau^2 I)$ em que $\boldsymbol{X}$ é a matriz $n \times p$ de covariáveis, $\boldsymbol{\beta}$ é o vetor de coeficientes, e $\boldsymbol{\Sigma}$ depende dos parâmetros (ϕ, σ^2, ν) da função de covariância Matérn (Diggle et al., 2007). A função de log-verossimilhança é:

$$\ell(\boldsymbol{\beta}, \theta) = -\frac{1}{2}\Big\{\log|\mathbf{V}| + (\mathbf{z} - \boldsymbol{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{z} - \boldsymbol{X}\boldsymbol{\beta})\Big\}, \tag{2.7}$$

na qual $\mathbf{V} = \boldsymbol{\Sigma} + \tau^2 I$ e cuja maximização resulta nas estimativas de máxima verossimilhança dos parâmetros do modelo.

3 GNN

A rede neural de grafos (GNN) é um tipo dec rede neural construída para trabalhar com dados representados por grafos. Em sua essência, as GNNs aplicam o poder preditivo do deep learning a dados que tem estrutura de dependência que descrevem um conjunto de objetos (nós) e suas relações (arestas) (Scarselli et al., 2008). GNNs são excelentes na captação de relações e ligações entre entidades, o que as torna ideais para tarefas como a análise de tráfego, sistemas de recomendação, modelação molecular e dados espaciais, entre alguns exemplos temos os seguintes artigos: Jiang and Luo (2022); Li et al. (2023); Wieder et al. (2020).

3.1 Grafo

Um grafo é uma estrutura matemática feita com o objetivo de analisar um conjunto de objetos e as relações deles entre si. Grafos são utilizados em múltiplas áreas, como na computação para representar problemas de otimização como em Diestel (2025). A área foi criada originalmente em 1736 quando Leonhard Euler mostrou a solução negativa do problema das Sete pontes de Königsberg, onde era discutido se podia cruzar todas as 7 pontes sem repetir nenhuma.

A estrutura fundamental na teoria é o denominado grafo simples. Formalmente, um grafo simples G é definido pelo par $G = (V, E)$, em que V é um conjunto finito de elementos chamados vértices (ou nodos) e E é um conjunto de pares não ordenados de elementos distintos de V, denominados arestas (Williamson, 2010). As Figuras (3.1,3.2) ilustram o clássico problema das Sete Pontes de Königsberg juntamente com sua representação em um grafo. Nesta modelagem, observa-se um sistema composto por 4 vértices e 7 arestas.

Pode-se observar na Figura 3.2 que os vértices (A,B) são adjacentes pela existência de uma aresta entre eles.

3.1.1 Representação Matricial

Embora seja conveniente representar grafos como na Figura 3.2, isso é apenas uma representação visual e matematicamente ele deve ser representado por matrizes Zhang and Chartrand (2006).

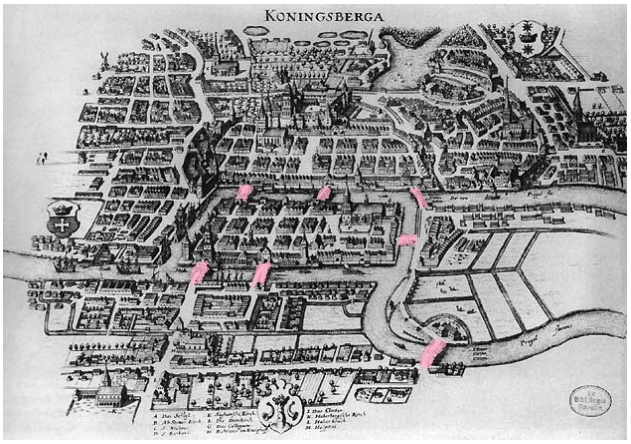

Figura 3.1: Pintura das 7 pontes de Königsberg.
Fonte: von Merian-Erben (1652), pontes destacadas pela autor

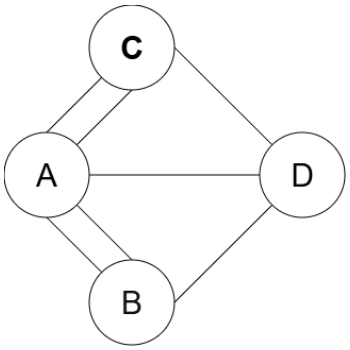

Figura 3.2: Grafo das sete pontes.

- **Matriz de Adjacência** – é a matriz quadrada de ordem n (número de vértices) onde a_{ij} representa o número de arestas conectando os vértices i e j.

$$\mathbf{A} = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 0 & 2 & 1 \\ B & 0 & 0 & 2 & 1 \\ C & 2 & 2 & 0 & 1 \\ D & 1 & 1 & 1 & 0 \end{array}$$

- **Matriz de Incidência** – é a matriz $n \times m$ (número de arestas), podemos ver na Matriz B abaixo que $b_{ij} = 1$ se o vértice i é uma extremidade da aresta j.

$$\mathbf{B} = \begin{array}{c|ccccccc} & a & b & c & d & e & f & g \\ \hline A & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ C & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ D & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}$$

Uma matriz de adjacência pode ser considerada um caso especial de uma matriz de vizinhança. A grande diferença entre elas é que a matriz de vizinhança frequentemente incorpora informações sobre a intensidade dessa relação. Isso será relevante no Capítulo 4.3.

3.2 Graph Embedding

Diferente de outros tipos de dados, como imagens ou texto, os grafos possuem uma estrutura de dados não-euclidiana. Eles não têm uma ordem espacial fixa, e a vizinhança de cada nó é variável. Redes neurais tradicionais, como CNNs ou MLPs, exigem que a entrada seja um vetor ou tensor de tamanho fixo. Elas não conseguem processar diretamente uma estrutura de grafo, como uma matriz de adjacência, de forma a capturar eficientemente suas propriedades topológicas Bronstein et al. (2017).

Para superar essa incompatibilidade, o primeiro passo para trabalhar com grafos é encontrar uma maneira de representar os componentes do grafo em um formato que a rede neural possa entender: um vetor numérico. Esse processo de tradução de estrutura em vetores é conhecido como embedding.

Graph embedding é um meio de representar um grafo como vetores de baixa dimensionalidade mantendo preservada sua estrutura original (Cai et al. (2018)). Um embedding de grafos tem o princípio de representar as covariáveis, matriz de adjacência ou o grafo inteiro como um campo vetorial. A formação de um embedding depende da técnica usada para construí-lo, que está diretamente ligada ao tipo de rede neural de grafos usada e o que ele representa está ligado aos níveis de predição.

3.3 Níveis de Predição

Redes neurais de grafo podem trabalhar com 3 níveis de predição diferente: predição em nível de nó, nível de aresta e nível de grafo (node embedding, edge embedding and whole-graph embedding)Sanchez-Lengeling et al. (2021a). Isso é ilustrado pela Figura 3.4, não serão comentados embeddings de subníveis (substructure embedding).

- Nível de nó: preocupam-se em prever a identidade ou o papel de cada nó dentro de um grafo. Esta é uma das tarefas mais fundamentais e é diretamente análoga aos problemas clássicos de interpolação e classificação espacial. Em uma GNN, isso é frequentemente chamado de classificação de nó ou regressão de nó.

- Nível de Aresta: é aplicado quando se quer prever quais nós compartilham uma aresta ou qual é o valor dessa aresta. Se o objetivo é descobrir conexões entre entidades, pode-se considerar o grafo como totalmente conectado e, com base no valor previsto, podar as arestas para se chegar a um grafo esparso.

- Nível de Grafo: o objetivo é predizer as propriedades do grafo inteiro. Isso é análogo a problemas de classificação de imagem na qual se associa um rótulo com uma imagem inteira. Esse tipo de técnica pode ser aplicado em estudos moleculares para a predição de suas características. Han et al. (2022) usou uma arquitetura de GNN chamada Vision GNN que transformou imagens em grafos com um grupo de pixeis como nodos e as classificou.

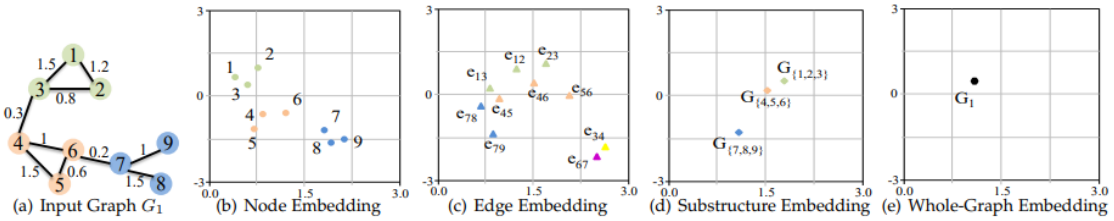

Figura 3.3: diferente tipos de embeddings de um grafo.
Fonte: Cai et al. (2018)

3.4 Camada de uma GNN

As camadas constituem o componente fundamental das GNNs, sendo responsáveis pela troca de informações entre nodos vizinhos e a atualização de seus embeddings. Diferentemente das Redes Neurais Perceptrons Multicamadas (MLP), onde a profundidade está primariamente associada ao nível de abstração das características, nas GNNs o número de camadas determina a expansão do campo receptivo do nodo alvo, ou seja, a quantos níveis de vizinhança a informação é capturada. Embora exista uma vasta diversidade de formulações na literatura, Bronstein et al. (2021) propõem uma taxonomia baseada na hierarquia de expressividade, organizando as três principais arquiteturas (Convolucional, Atencional e Message-Passing) através de uma relação de contenção:

$$\text{Convolution} \subseteq \text{Attention} \subseteq \text{Message-Passing},$$

nesta estrutura, o Message-Passing apresenta-se como o mecanismo mais genérico de agregação e atualização, do qual as camadas de atenção e convolucionais são casos particulares ou restritos. Todas essas abordagens compartilham o uso de funções com pesos aprendíveis e a propriedade de invariância à permutação, assegurando que o modelo produza o mesmo resultado independentemente da ordem em que os vizinhos são processados pela função de agregação, conforme será detalhado nos itens a seguir.

- **Message Passing**

  Seja um grafo uma tupla de nós e arestas, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, com vizinhanças de um salto definidas como $\mathcal{N}_i = \{k \in \mathcal{V} | (v, i) \in \mathcal{E}\}$ Bronstein et al. (2021). Além disso, uma matriz de características de nó $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times k}$ fornece as características do nó u como $\mathbf{x}_u$. Em nível de nodo, sem levar em conta características de nível de aresta e do grafo, uma GNN de troca de mensagens sobre este grafo é então executada como:

$$\mathbf{g}_i = \Upsilon\left(\mathbf{q}_i, \bigoplus_{k \in \mathcal{N}_i} \psi(\mathbf{q}_i, \mathbf{q}_k)\right),$$

  na qual $\psi : \mathbb{R}^v \times \mathbb{R}^v \to \mathbb{R}^l$ é uma função de mensagem, $\Upsilon : \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}^m$ é uma função de leitura e $\bigoplus$ é uma função de agregação invariante à permutação como somatório ou média. Tanto ψ quanto Υ podem ser implementadas de formas diversas sendo definidas de acordo com o tipo de GNN usada, podendo ser definida como uma função linear ou uma MLP $\mathbf{d}^{(l)} = \sigma\left(\mathbf{W}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}\right)$,

mas existem muitos casos especiais, dando origem, por exemplo, às GNNs atencionais.

- **Atencional**

  Velickovic et al. (2017) Definiu a GAT como um mecanismo de auto-atenção aprendível que calcula os coeficientes de importância $\alpha_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$ que são frequentemente normalizados via softmax entre todos os vizinhos.

$$\mathbf{h}_u = \Upsilon\left(\mathbf{q}_u, \bigoplus_{v \in \mathcal{N}_u} a(\mathbf{q}_u, \mathbf{q}_v)\psi(\mathbf{q}_v)\right),$$

  Vale a pena dizer que, embora similar, o sistema de atenção de uma GAT é diferente do que o escrito por Vaswani et al. (2017) em um transformer. A atenção aplicada aqui leva em conta apenas a matriz de vizinhnaça, diferente do transformer que tem um sistema de atenção global.

- **Convolucional**

  Na abordagem convolucional Sanchez-Lengeling et al. (2021b), as características dos nós da vizinhança são agregadas diretamente com pesos fixos. Aqui, c_{uv} especifica a importância do nó v para a representação do nó u.

$$\mathbf{h}_u = \Upsilon\left(\mathbf{q}_u, \bigoplus_{v \in \mathcal{N}_u} c_{uv}\psi(\mathbf{q}_v)\right).$$

3.5 Arquitetura

Em comparação com as redes neurais clássicas, a GNN tem um número de camadas ocultas igual à profundidade da vizinhança; isso faz com que ela tenha consideravelmente menos camadas (Message Passing Layer, MPL) do que as redes normais. Isso pode ser visto na Figura 3.4, em que: (a) é o grafo de entrada (input graph) com o nó alvo B; (b) é o fluxo dos nós (nodeflow), que mostra qual nó é vizinho e enviará mensagem para outro nó em cada camada; e (c) é o fluxo de dados de inferência, que decompõe uma camada nos passos de função de mensagem, agregação e resposta (atualização).

3.6 Treinamento

Em GNN, é fundamental distinguir entre inferência transdutiva e indutiva, justamente pela diferença de metodologia no treinamento. Formalmente, a diferença reside no objeto matemático que a rede neural busca aproximar durante o treinamento Li et al. (2024).

A abordagem transdutiva não busca necessariamente uma regra geral, mas sim mapear os dados de treino e as entradas não rotuladas diretamente para as predições daquele conjunto de teste específico, nesse caso, a rede recebe um grafo com todas as arestas, mas os rótulos não são mostrados. Em contraste, a inferência indutiva foca em aprender uma função de hipótese $f : \mathcal{X} \to \mathcal{Z}$ capaz de generalizar para

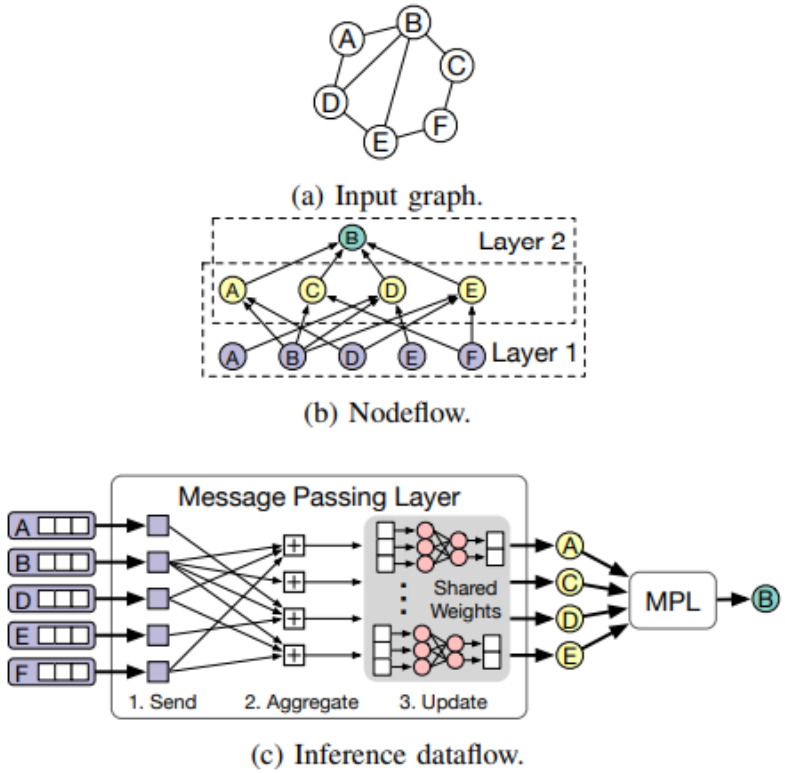

Figura 3.4: Arquitetura de uma GNN.

Fonte: Cai et al. (2018)

qualquer entrada futura, nessa caso a rede vê apenas um subgrafo no treinamento, tornando o modelo independente dos dados de treino após a fase de aprendizado.

3.7 Desafios da Estimação na GNN

Dois fenômenos comumente observados em redes que utilizam técnicas análogas ao *message-passing* são a super-suavização (over-smoothing) e o super-achatamento (over-squashing).

Super-suavização é resultado das características dos nós em GNNs tenderem a se tornar mais similares à medida que a profundidade da rede aumenta. Rusch et al. (2023) definiu isso como a convergência exponencial de medidas de similaridade sobre as características dos nós.

Super-achatamento é um fenômeno que se manifesta em tarefas que exigem a integração de informações de nós distantes Akansha (2025). Ele denota a natureza propensa à distorção da transferência de informação entre nós que estão muito separados devido à tensão inerente entre a capacidade limitada de representação de características dos embeddings do grafo e o crescimento exponencial no número de vizinhos à medida que os grafos se expandem.

4 NN-GLS

Como vimos, os Processos Gaussianos (GP) oferecem uma estrutura estatística robusta e uma quantificação de incerteza rigorosa. No entanto, sua principal desvantagem, além do número de suposições, é o alto custo computacional, que escala cubicamente $O(n^3)$ com o número de observações, tornando-os proibitivos para os grandes bases de dados Lin et al. (2025). Por outro lado, as Redes Neurais de Grafos (GNNs) apresentam excelente escalabilidade e flexibilidade para aprender padrões complexos em dados estruturados espacialmente. Contudo, elas não incorporam nativamente os princípios fundamentais da geoestatística, como a estrutura de covariância que é central nos Processos Gaussianos.

No artigo Zhan and Datta (2025), referência de todo esse capítulo, os autores propuseram a NN-GLS (Neural Network-Generalized Least Squares), com a intenção de utilizar em conjunto a não linearidade de redes e a dependência espacial, que normalmente é descartada quando utiliza-se redes neurais. Para isso, a NN-GLS incorpora a não-linearidade na média através da estimação, enquanto o GP leva em conta a dependência espacial mantendo as vantagens da modelagem através da covariância espacial e na predição de pontos não observados através da krigagem.

4.1 Fundamentos Iniciais

4.1.1 OLS x GLS

A motivação do algoritmo fundamenta-se na extensão dos Mínimos Quadrados Ordinários (OLS) para os Mínimos Quadrados Generalizados (GLS), conforme a Equação 2.6, para modelos lineares com erros dependentes. Segundo (Atanlogun et al., 2014) o GLS é aplicado quando as variâncias das observações são desiguais (heterocedasticidade) ou quando há correlação entre as observações. Nesses casos, o OLS pode ser estatisticamente ineficiente, ou até mesmo gerar inferências enganosas.

4.2 Modelo teórico

O modelo é definido como uma expansão do modelo de krigagem de forma que:

$$Z(s) = f(\mathbf{X}(s)) + \omega(s); \quad \omega(\cdot) \sim GP(0, \mathbf{\Sigma}).$$

Seja $\mathbf{f}(\mathbf{X}) = (f(\mathbf{X}_1), f(\mathbf{X}_2), \ldots, f(\mathbf{X}_n))^\top$. Então a verossimilhança é baseada na

seguinte distribuição de probabilidade:

$$\mathbf{Z} \sim \mathcal{N}(\mathbf{f}(\mathbf{X}), \mathbf{\Sigma}).$$

Esta é uma generalização não-linear do modelo linear espacial (Equação **??**). Para a estimação de parâmetros em modelos lineares para dados dependentes, a perda OLS é substituída por uma perda de GLS usando a matriz de covariância $\mathbf{\Sigma}$, pois é mais eficiente de acordo com o teorema de Gauss-Markov. Similarmente, para estimar f usando redes neurais, é proposta a seguinte função perda:

$$\mathcal{L}_n(f) = \frac{1}{n}(\mathbf{Z} - \mathbf{f}(\mathbf{X}))^\top \mathbf{Q}(\mathbf{Z} - \mathbf{f}(\mathbf{X})),$$

que leva em conta a dependência espacial através da matriz de precisão $\mathbf{Q}$, que é igual a $\mathbf{\Sigma}^{-1}$ ou, mais realisticamente, uma estimativa dela.

Em geral, a estimação direta desse método é incomputável, visto que ele gera inúmeros problemas computacionais. Por exemplo, minilotes não podem ser aplicados para uma perda GLS, a retropropagação requer a inversão de uma matriz densa Σ de tamanho $(n \times n)$ que requer $O(n^2)$ de armazenamento e $O(n^3)$ de tempo para cada iteração. Essas necessidades computacionais são inviáveis mesmo para um n moderado. A solução é utilizar um algoritmo que representa esse modelo como uma espécie de GNN Zhan and Datta (2025).

4.3 NN-GLS como uma GNN

Para lidar com a alta complexidade computacional, são tomadas 3 ações principais:

- aplicação do método Processo Gaussiano de Vizinhos Mais Próximos (NNGP), proposto por Datta et al. (2016);
- uma decorrelação aos dados;
- adaptação da estrutura da rede neural normal para uma rede neural de grafos.

O método NNGP é usado para modelar o efeito aleatório estruturado espacialmente quando o número de localizações espaciais n é grande. O NNGP fornece uma aproximação esparsa $\tilde{\mathbf{\Sigma}}^{-1}$ para a matriz de precisão densa completa do GP $\mathbf{\Sigma}^{-1}$. Isso é construído usando um grafo acíclico direcionado (DAG) baseado em distâncias par a par entre as n localizações de dados, de modo que cada nó tenha no máximo $m \ll n$ vizinhos direcionados.

Sendo $N(i)$ o conjunto de vizinhos da localização s_i no DAG, temos:

$$\tilde{\mathbf{\Sigma}}^{-1} = (\mathbf{I} - \mathbf{B})^\top \mathbf{F}^{-1}(\mathbf{I} - \mathbf{B}), \tag{4.1}$$

$$\mathbf{B}_{i,N(i)} = \mathbf{\Sigma}_{i,N(i)}\mathbf{\Sigma}^{-1}_{N(i),N(i)}, \quad B_{ij} = 0 \text{ em outros casos}, \tag{4.2}$$

$$\mathbf{F}_{ii} = \mathbf{\Sigma}_{ii} - \mathbf{\Sigma}_{i,N(i)}\mathbf{\Sigma}^{-1}_{N(i),N(i)}\mathbf{\Sigma}_{N(i),i}. \tag{4.3}$$

em que $\mathbf{B}$ é uma matriz estritamente triangular inferior e $\mathbf{F}$ é uma matriz diagonal, essas matrizes também denotam, respectivamente, os pesos da krigagem para predizer Z_0 e à variância da krigagem (variância condicional). As matrizes de precisão do modelo NNGP exigem apenas a inversão de n pequenas matrizes de tamanho

$(m \times m)$. Como $m \ll n$, o cálculo da verossimilhança requer um tempo e armazenamento total de $O(n)$.

A perda GLS pode ser vista como uma perda OLS para a resposta decorrelacionada $\mathbf{Z}^* = \mathbf{Q}^{1/2}\mathbf{Z}$, onde $\mathbf{Q}^{1/2}$ é o fator de Cholesky tal que $\mathbf{Q} = (\mathbf{Q}^{1/2})^\top \mathbf{Q}^{1/2}$. Logo, a decorrelação é simplesmente uma operação linear. Ao escolher $\mathbf{Q} = \tilde{\mathbf{\Sigma}}^{-1}$, a decorrelação se torna uma convolução no DAG de vizinhos mais próximos.

Seja $N^*[i] = N(i) \cup \{i\}$ a vizinhança do grafo para o i-ésimo nó. Definindo os pesos como:

$$\mathbf{\Xi}_i^\top = \frac{1}{\sqrt{\mathbf{F}_{ii}}}(1, -\mathbf{B}_{i,N(i)}),$$

a resposta decorrelacionada torna-se $Z_i^* = \mathbf{\Xi}_i^\top \mathbf{Z}_{N^*[i]}$. Os conjuntos de índices $N(i)$ ou $N^*[i]$ são usados no subscrito para criar subconjuntos de uma matriz ou de um vetor. Assim, as respostas decorrelacionadas Z_i^* são simplesmente uma convolução sobre o DAG utilizado no NNGP, com os pesos de convolução de grafo $\mathbf{\Xi}_i$ definidos usando krigagem. Similarmente, pode-se definir a camada de saída decorrelacionada $o_i^* = \mathbf{\Xi}_i^\top \mathbf{o}_{N^*[i]}$ usando a mesma convolução de grafo, onde $\mathbf{o}_i$ representa os valores da camada de saída da rede neural f.

4.4 Arquitetura

A Figura 4.1 detalha o funcionamento dessa arquitetura. Primeiramente, as covariáveis $\mathbf{X}_i$ e as respostas $\mathbf{Z}_i$ são estruturadas como um DAG, canto superior da figura. Após isso, no diagrama central, as covariáveis $\mathbf{X}_i$ são processadas por uma rede neural MLP para produzir uma saída $\mathbf{O}_i = f(\mathbf{X}_i)$. Esta é uma operação intra-nó, permitindo o uso de qualquer arquitetura (número de camadas, neurônios, esparsidade ou funções de ativação).

Em seguida, os valores de saída O_i da MLP são submetidos a uma convolução de grafo adicional para gerar a saída decorrelacionada O_i^*, nodo roxo na figura, utilizando os pesos $\mathbf{\Xi}_i$. Este valor é comparado, via perda OLS, com a resposta decorrelacionada Z_i^*, obtida a partir das observações originais Z_i através da mesma operação de convolução, visto na parte superior do diagrama central. Assim, a função objetivo é expressa como:

$$\sum_i (Z_i^* - O_i^*)^2, \quad \text{onde } Z_i^* = \mathbf{\Xi}_i^\top \mathbf{Z}_{N^*[i]} \quad \text{e} \quad O_i^* = \mathbf{\Xi}_i^\top \mathbf{O}_{N^*[i]}.$$

4.5 Treinamento

Considerando a autocorrelação dos dados e a necessidade de preservar a estrutura de vizinhança do NNGP, não é viável particionar os dados aleatoriamente. Portanto a rede utiliza o método de divisão baseada em blocos. O domínio é dividido em uma grade $K \times K$, da qual K blocos são selecionados para o banco de teste. A seleção segue uma restrição geométrica onde apenas um bloco é escolhido por linha e por coluna nas quais o domínio observado é dividido, buscando representatividade espacial e independência estatística entre os conjuntos.

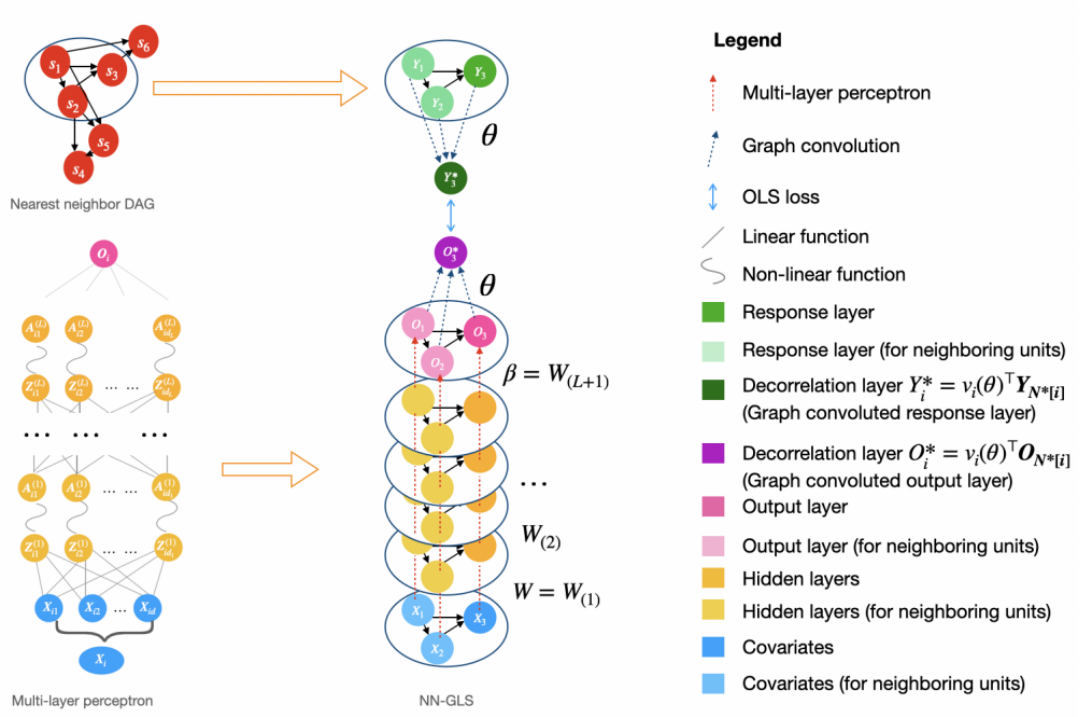

Figura 4.1: Arquitetura da NN-GLS.

Fonte: Zhan and Datta (2025).

4.6 Krigagem

Conforme o artigo Zhan and Datta (2025), para fazer a predição de uma conjunto de novas localizações s_0, sendo $\mathbf{Z_0} := \mathbf{Z}(s_0)$ o vetor de pontos a serem preditos. O vetor $\mathbf{Z_0}$ é inserido como input na MLP treinada e tem como output $O_0 = f(\mathbf{X_0})$. Com isso, a nova localização s_0 é inserida no DAG, $N^*[0] = \{s_0\} \cup N(0)$, é feita a mesma etapa de $O_0^* = v_0 O_{N[0]}$ que é o modelo para a camada de resposta decorrelacionada Z_0^* que nesse caso é $\hat{Z}_0 = O_0$. Após isso precisamos desconvolucionar essa camada de forma:

$$\hat{Z}_0 = \sqrt{\mathbf{F}_{00}}\hat{Z}_0^* + \mathbf{B}_{0,N(0)}^\top \mathbf{Z}_{N(0)},$$

que também pode ser representada como:

$$\hat{Z}_0 = \hat{f}(\boldsymbol{X}_0) + \boldsymbol{\Sigma}(s_0, N(0))\boldsymbol{\Sigma}(N(0), N(0))^{-1}(\boldsymbol{Z}_{N(0)} - \hat{\boldsymbol{f}}_{N(0)}).$$

Além disso, a variância de predição é simplesmente $\sigma_0^2 = \mathbf{F}_{00}$. Como a NN-GLS está inserida na estrutura de Processos Gaussianos, as distribuições preditivas condicionais aos dados e parâmetros são Gaussianas.

5 Simulação

Para avaliar o desempenho do EMV e da NN-GLS em cenários espaciais controlados, os métodos foram aplicados em dois conjuntos de dados simulados. O objetivo das simulações é testar a capacidade do NN-GLS e do EMV de lidar com diferentes estruturas de média e diferentes parâmetros.

5.1 Pré-Processamento dos Dados

Em todos os cenários foi usado 80% do banco para treinamento, 10% para a validação e 10% como teste. No caso da krigagem, os bancos de treinamento e validação foram usados na estimação dos parâmetros com a função Likfit do pacote geoR Ribeiro Jr and Diggle (2025). Todos os dados foram normalizados para as redes neurais.

5.2 Estudo de Simulações

5.2.1 Cenário 1

Para as simulações foi utilizada a função `grf` do pacote `geoR`, que geraram campos aleatórios gaussianos como os apresentados na Figura 5.1. A krigagem foi calculada utilizando o mesmo pacote no ambiente R (versão 4.3.2), enquanto a implementação do método NN-GLS foi conduzida em Python (versão 3.10) através da biblioteca geospaNN (Zhan and Datta, 2025).

O estudo experimental compreende dois conjuntos de simulação. O primeiro é constituído por 48 bases de dados, cada uma contendo 400 observações distribuídas em localizações aleatórias geradas por duas distribuições uniformes $U[0, 10]$. A componente determinística da média (μ) é modelada em função de três covariáveis, denotadas por X_1, X_2 e X_3, cujos coeficientes de regressão são fixados em $\beta_1 = 1.5$, $\beta_2 = -0.8$ e $\beta_3 = 1.2$, respectivamente. As covariáveis seguem as distribuições uniformes $X_1 \sim U[1, 2]$, X_2 e $X_3 \sim U[0, 1]$, gerando médias que obedecem ao seguinte padrão:

- Constante: $\mu(s) = 5$;
- Linear: $\mu(s) = \beta_1 x_1(s) + \beta_2 x_2(s) + \beta_3 x_3(s)$;
- Complexa: $\mu(s) = log(x_1(s)) + sin(x_2(s)) + x_3(s)^2$.

A simulação também levou em conta diferentes valores de $\sigma^2 = (1, 3, 5, 10)$, $\phi = (1, 5, 8, 10)$ e $\tau^2 = 1$ para todas as simulações.

Para simulação de Monte Carlo foram consideradas 400 repetições para cada uma dessas combinações.

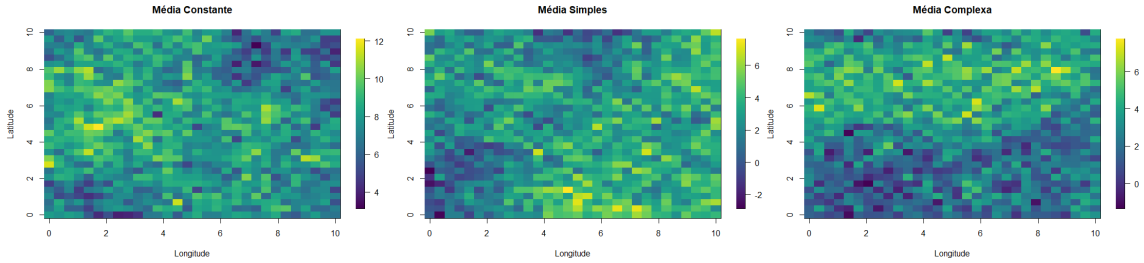

Figura 5.1: Exemplos de diferentes campos gaussianos simulados com parâmetros para a função de covariância $\sigma^2 = 3$ e $\phi = 5$, mas diferentes estruturas de média: Constante, Linear e Complexa.

5.2.2 Comparação dos Resultados - Cenário 1

A Tabelas 5.1 e 5.4 indicam as medianas, o primeiro e o terceiro quartis dos parâmetros estimados para σ^2 (patamar), ϕ (alcance) e τ^2 (efeito pepita) das 400 simulações de Monte Carlo.

Tabela 5.1: Cenário 1: estimativas dos parâmetros considerando $\sigma^2 = 1$, $\tau^2 = 1$ e $phi = (1, 5, 8, 10)$.

	σ^2 (Patamar)		ϕ (Alcance)		τ^2 (Nugget)	
	EMV	NN-GLS	EMV	NN-GLS	EMV	NN-GLS
$\phi = 1$						
Constante	0.980 (0.825, 1.129)	1.015 (0.778, 1.261)	0.894 (0.740, 1.119)	0.870 (0.644, 1.179)	0.962 (0.857, 1.064)	1.256 (0.897, 1.790)
Linear	0.978 (0.829, 1.162)	0.973 (0.793, 1.198)	0.901 (0.713, 1.118)	0.877 (0.665, 1.175)	0.957 (0.841, 1.069)	1.186 (0.798, 1.541)
Complexa	0.987 (0.835, 1.169)	0.991 (0.815, 1.202)	0.916 (0.725, 1.146)	0.879 (0.669, 1.196)	0.969 (0.852, 1.067)	1.045 (0.790, 1.375)
$\phi = 5$						
Constante	0.659 (0.477, 0.897)	0.649 (0.481, 0.887)	2.548 (1.776, 4.011)	2.383 (1.485, 4.268)	0.972 (0.903, 1.039)	1.957 (1.386, 2.694)
Linear	0.636 (0.454, 0.906)	0.608 (0.417, 0.856)	2.678 (1.834, 4.163)	2.527 (1.640, 4.006)	0.970 (0.906, 1.038)	1.852 (1.235, 2.720)
Complexa	0.650 (0.470, 0.898)	0.626 (0.445, 0.843)	2.619 (1.762, 4.019)	2.490 (1.598, 3.983)	0.968 (0.896, 1.036)	1.616 (1.135, 2.382)
$\phi = 8$						
Constante	0.498 (0.348, 0.716)	0.452 (0.300, 0.671)	2.999 (1.891, 4.896)	3.110 (1.757, 5.550)	0.974 (0.911, 1.033)	2.814 (1.863, 4.341)
Linear	0.490 (0.328, 0.728)	0.478 (0.321, 0.663)	3.029 (1.781, 5.096)	2.933 (1.656, 5.155)	0.976 (0.917, 1.035)	2.233 (1.619, 3.708)
Complexa	0.492 (0.347, 0.730)	0.477 (0.320, 0.667)	3.271 (2.003, 5.064)	3.088 (1.740, 5.241)	0.978 (0.924, 1.034)	2.189 (1.564, 3.395)
$\phi = 10$						
Constante	0.435 (0.307, 0.672)	0.431 (0.298, 0.645)	3.279 (2.123, 5.172)	3.087 (1.621, 5.755)	0.960 (0.901, 1.022)	2.898 (1.973, 4.361)
Linear	0.428 (0.296, 0.620)	0.387 (0.272, 0.549)	3.305 (1.921, 5.619)	3.501 (1.877, 6.800)	0.969 (0.911, 1.025)	2.952 (1.996, 4.363)
Complexa	0.445 (0.302, 0.667)	0.402 (0.272, 0.584)	3.343 (2.138, 5.648)	2.999 (1.730, 5.905)	0.980 (0.918, 1.033)	2.600 (1.723, 3.994)

Tabela 5.2: Resumo Comparativo de Estimação - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 3$).

	σ^2 (Patamar)		ϕ (Alcance)		τ^2 (Nugget)	
	EMV	NN-GLS	EMV	NN-GLS	EMV	NN-GLS
$\phi = 1$						
Constante	2.916 (2.591, 3.301)	3.040 (2.603, 3.493)	0.921 (0.774, 1.089)	0.884 (0.705, 1.089)	0.934 (0.781, 1.096)	0.386 (0.289, 0.528)
Linear	2.884 (2.606, 3.322)	2.980 (2.556, 3.344)	0.882 (0.750, 1.076)	0.896 (0.734, 1.078)	0.930 (0.768, 1.076)	0.372 (0.267, 0.478)
Complexa	2.875 (2.542, 3.292)	2.881 (2.478, 3.327)	0.909 (0.753, 1.074)	0.915 (0.745, 1.130)	0.956 (0.815, 1.107)	0.362 (0.274, 0.478)
$\phi = 5$						
Constante	1.981 (1.461, 2.800)	1.904 (1.407, 2.702)	3.104 (2.166, 4.486)	3.115 (2.037, 4.641)	0.964 (0.889, 1.038)	0.653 (0.443, 0.944)
Linear	2.015 (1.447, 2.662)	1.974 (1.489, 2.581)	3.084 (2.067, 4.327)	2.916 (1.995, 3.946)	0.960 (0.887, 1.033)	0.562 (0.422, 0.773)
Complexa	1.917 (1.444, 2.612)	1.873 (1.391, 2.562)	2.918 (2.057, 4.261)	2.802 (1.916, 4.275)	0.968 (0.895, 1.040)	0.550 (0.394, 0.750)
$\phi = 8$						
Constante	1.530 (1.076, 2.282)	1.499 (1.051, 2.140)	3.704 (2.408, 5.541)	3.486 (2.197, 5.077)	0.964 (0.886, 1.036)	0.851 (0.568, 1.240)
Linear	1.496 (1.054, 2.164)	1.428 (0.986, 2.056)	3.556 (2.478, 5.799)	3.340 (2.178, 5.589)	0.965 (0.905, 1.039)	0.785 (0.505, 1.151)
Complexa	1.437 (1.010, 2.177)	1.392 (0.939, 2.120)	3.466 (2.290, 5.798)	3.350 (2.090, 5.235)	0.967 (0.897, 1.047)	0.746 (0.480, 1.130)
$\phi = 10$						
Constante	1.245 (0.862, 1.888)	1.162 (0.834, 1.735)	3.493 (2.275, 5.922)	3.652 (2.266, 5.994)	0.958 (0.899, 1.021)	1.054 (0.685, 1.554)
Linear	1.249 (0.856, 1.972)	1.196 (0.828, 1.802)	3.507 (2.286, 6.173)	3.480 (2.170, 5.853)	0.971 (0.911, 1.032)	0.926 (0.634, 1.355)
Complexa	1.319 (0.894, 2.040)	1.260 (0.891, 1.869)	3.757 (2.367, 6.068)	3.516 (2.249, 5.758)	0.970 (0.897, 1.033)	0.847 (0.526, 1.245)

Tabela 5.3: Resumo Comparativo de Estimação - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 5$).

	σ^2 (Patamar)		ϕ (Alcance)		τ^2 (Nugget)	
	EMV	NN-GLS	EMV	NN-GLS	EMV	NN-GLS
$\phi = 1$						
Constante	4.791 (4.285, 5.406)	4.795 (4.193, 5.400)	0.938 (0.804, 1.103)	0.946 (0.789, 1.158)	0.962 (0.788, 1.151)	0.267 (0.212, 0.356)
Linear	4.739 (4.174, 5.464)	4.892 (4.243, 5.596)	0.936 (0.785, 1.132)	0.899 (0.755, 1.102)	0.922 (0.735, 1.113)	0.209 (0.161, 0.287)
Complexa	4.737 (4.263, 5.270)	4.745 (4.157, 5.419)	0.919 (0.765, 1.097)	0.887 (0.740, 1.109)	0.945 (0.739, 1.152)	0.215 (0.157, 0.291)
$\phi = 5$						
Constante	3.482 (2.467, 4.710)	3.299 (2.413, 4.699)	3.252 (2.214, 4.615)	2.874 (2.008, 4.508)	0.971 (0.865, 1.056)	0.369 (0.260, 0.533)
Linear	3.233 (2.316, 4.384)	3.178 (2.313, 4.309)	2.857 (2.048, 4.197)	2.780 (1.869, 4.287)	0.956 (0.858, 1.036)	0.338 (0.253, 0.486)
Complexa	3.278 (2.477, 4.598)	3.222 (2.494, 4.497)	3.175 (2.147, 4.456)	2.948 (2.020, 4.383)	0.970 (0.888, 1.055)	0.315 (0.226, 0.438)
$\phi = 8$						
Constante	2.467 (1.806, 3.498)	2.371 (1.749, 3.493)	3.415 (2.338, 5.266)	3.382 (2.305, 5.258)	0.949 (0.873, 1.027)	0.535 (0.349, 0.779)
Linear	2.462 (1.814, 3.927)	2.512 (1.830, 3.605)	3.655 (2.451, 6.216)	3.582 (2.299, 5.450)	0.961 (0.892, 1.032)	0.439 (0.305, 0.598)
Complexa	2.400 (1.807, 3.739)	2.349 (1.789, 3.514)	3.461 (2.498, 5.641)	3.302 (2.316, 5.357)	0.976 (0.895, 1.052)	0.439 (0.282, 0.623)
$\phi = 10$						
Constante	2.135 (1.465, 3.210)	2.087 (1.438, 2.978)	3.679 (2.483, 6.124)	3.706 (2.351, 5.897)	0.955 (0.887, 1.035)	0.613 (0.408, 0.953)
Linear	2.094 (1.440, 3.169)	2.014 (1.379, 2.960)	3.751 (2.451, 5.883)	3.690 (2.233, 5.626)	0.956 (0.886, 1.033)	0.566 (0.378, 0.825)
Complexa	2.211 (1.527, 3.417)	2.149 (1.456, 3.377)	4.059 (2.429, 6.308)	3.567 (2.288, 5.947)	0.969 (0.892, 1.045)	0.487 (0.325, 0.710)

Tabela 5.4: Resumo Comparativo de Estimação - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 10$).

	σ^2 (Patamar)		ϕ (Alcance)		τ^2 (Nugget)	
	EMV	NN-GLS	EMV	NN-GLS	EMV	NN-GLS
$\phi = 1$						
Constante	9.498 (8.437, 10.720)	9.492 (8.471, 10.732)	0.914 (0.781, 1.060)	0.889 (0.755, 1.075)	0.848 (0.593, 1.136)	0.121 (0.077, 0.187)
Linear	9.696 (8.585, 10.968)	9.714 (8.606, 10.946)	0.943 (0.812, 1.087)	0.922 (0.790, 1.069)	0.879 (0.649, 1.158)	0.104 (0.070, 0.145)
Complexa	9.703 (8.570, 10.782)	9.767 (8.645, 10.833)	0.926 (0.792, 1.086)	0.920 (0.776, 1.115)	0.893 (0.625, 1.179)	0.101 (0.061, 0.142)
$\phi = 5$						
Constante	6.495 (4.976, 9.198)	6.402 (4.893, 9.012)	3.106 (2.207, 4.597)	2.929 (2.065, 4.290)	0.947 (0.824, 1.069)	0.192 (0.123, 0.273)
Linear	6.521 (4.904, 8.929)	6.494 (4.819, 8.964)	3.010 (2.199, 4.362)	3.039 (2.183, 4.394)	0.952 (0.847, 1.051)	0.178 (0.124, 0.244)
Complexa	6.627 (4.859, 9.430)	6.577 (4.907, 9.624)	3.124 (2.287, 4.712)	3.087 (2.174, 4.697)	0.950 (0.838, 1.064)	0.155 (0.110, 0.213)
$\phi = 8$						
Constante	4.732 (3.539, 7.429)	4.636 (3.417, 7.171)	3.552 (2.535, 5.813)	3.594 (2.454, 5.608)	0.955 (0.856, 1.051)	0.271 (0.176, 0.408)
Linear	5.078 (3.557, 7.595)	5.119 (3.629, 7.523)	3.729 (2.564, 5.725)	3.738 (2.492, 5.637)	0.949 (0.852, 1.041)	0.220 (0.155, 0.319)
Complexa	5.149 (3.625, 7.832)	5.126 (3.664, 7.790)	3.918 (2.720, 6.111)	3.926 (2.656, 5.701)	0.963 (0.875, 1.038)	0.217 (0.130, 0.299)
$\phi = 10$						
Constante	4.272 (3.018, 6.789)	4.329 (2.959, 6.829)	3.980 (2.667, 6.786)	4.112 (2.519, 6.440)	0.948 (0.861, 1.033)	0.300 (0.172, 0.454)
Linear	4.304 (3.064, 6.404)	4.192 (3.006, 5.905)	3.870 (2.623, 6.564)	3.725 (2.479, 5.399)	0.959 (0.871, 1.052)	0.265 (0.189, 0.396)
Complexa	4.503 (3.058, 6.734)	4.407 (2.973, 6.435)	4.205 (2.789, 6.672)	4.029 (2.548, 6.382)	0.963 (0.879, 1.046)	0.244 (0.166, 0.366)

A análise das Tabelas 5.1 a 5.4 revela padrões distintos de comportamento entre o EMV e a NN-GLS na estimação dos parâmetros de covariância, variando conforme a intensidade do patamar (σ^2) e o alcance (ϕ):

- Patamar (σ^2) e Alcance (ϕ): observa-se, em ambas as metodologias, uma subestimação crescente na estimação do patamar e do alcance à medida que o verdadeiro valor de ϕ aumenta.
- Efeito Pepita (τ^2): a diferença mais marcante entre os métodos reside na estimativa do efeito pepita. O EMV manteve as medianas das estimativas próximas ao valor real ($\tau^2 = 1$) em todos os cenários, enquanto a NN-GLS incorporou o erro não espacial com a estrutura de dependência em muitos casos, percebe-se que em todas as tabelas que o valor da estimação do parâmetro aumenta conforme os valores de ϕ.

A seguir são mostradas as Tabelas 5.5 a 5.8 que apresentam a mediana do erro-quadrático médio e os primeiro e terceiro quartil. O EQM consolida o viés e a variância da estimativa em uma única métrica de desempenho, permitindo avaliar a precisão global dos modelos.

Tabela 5.5: Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 1$).

	EQM (Erro Quadrático Médio)	
Parâmetro / Tendência	EMV	NN-GLS
$\phi = 1$		
Constante	0.650 (0.538, 0.801)	2.484 (1.886, 3.295)
Linear	0.782 (0.650, 0.961)	1.759 (1.510, 2.133)
Complexa	0.721 (0.583, 0.871)	1.644 (1.400, 1.908)
$\phi = 5$		
Constante	0.712 (0.589, 0.867)	1.553 (1.235, 2.114)
Linear	0.927 (0.777, 1.100)	1.322 (1.121, 1.550)
Complexa	0.806 (0.671, 0.989)	1.281 (1.109, 1.493)
$\phi = 8$		
Constante	0.765 (0.635, 0.905)	1.812 (1.309, 2.884)
Linear	1.015 (0.875, 1.185)	1.383 (1.127, 1.663)
Complexa	0.938 (0.779, 1.088)	1.258 (1.071, 1.475)
$\phi = 10$		
Constante	0.787 (0.664, 0.908)	1.840 (1.403, 3.205)
Linear	1.114 (0.944, 1.294)	1.315 (1.102, 1.636)
Complexa	0.868 (0.739, 1.012)	1.243 (1.041, 1.464)

Tabela 5.6: Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 3$).

	EQM (Erro Quadrático Médio)	
Parâmetro / Tendência	EMV	NN-GLS
$\phi = 1$		
Constante	0.606 (0.502, 0.771)	2.648 (2.227, 3.171)
Linear	0.637 (0.496, 0.757)	2.440 (2.098, 2.847)
Complexa	0.572 (0.451, 0.720)	2.377 (2.038, 2.747)
$\phi = 5$		
Constante	0.587 (0.494, 0.720)	1.517 (1.280, 1.770)
Linear	0.735 (0.608, 0.877)	1.503 (1.270, 1.746)
Complexa	0.693 (0.556, 0.834)	1.444 (1.227, 1.723)
$\phi = 8$		
Constante	0.614 (0.513, 0.764)	1.386 (1.164, 1.681)
Linear	0.941 (0.764, 1.097)	1.360 (1.164, 1.576)
Complexa	0.724 (0.591, 0.865)	1.341 (1.132, 1.588)
$\phi = 10$		
Constante	0.662 (0.532, 0.830)	1.403 (1.158, 1.677)
Linear	0.891 (0.759, 1.044)	1.354 (1.149, 1.545)
Complexa	0.759 (0.630, 0.925)	1.299 (1.102, 1.533)

Tabela 5.7: Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 5$).

	EQM (Erro Quadrático Médio)	
Parâmetro / Tendência	EMV	NN-GLS
$\phi = 1$		
Constante	0.570 (0.462, 0.703)	3.192 (2.787, 3.722)
Linear	0.587 (0.463, 0.738)	3.057 (2.650, 3.598)
Complexa	0.671 (0.501, 0.820)	3.015 (2.601, 3.621)
$\phi = 5$		
Constante	0.540 (0.441, 0.664)	1.659 (1.408, 1.972)
Linear	0.633 (0.519, 0.821)	1.617 (1.337, 1.881)
Complexa	0.572 (0.468, 0.713)	1.656 (1.398, 1.922)
$\phi = 8$		
Constante	0.594 (0.486, 0.704)	1.502 (1.257, 1.747)
Linear	0.728 (0.602, 0.865)	1.512 (1.263, 1.772)
Complexa	0.696 (0.578, 0.853)	1.477 (1.265, 1.706)
$\phi = 10$		
Constante	0.590 (0.481, 0.715)	1.426 (1.248, 1.677)
Linear	0.737 (0.602, 0.876)	1.429 (1.208, 1.685)
Complexa	0.659 (0.534, 0.814)	1.428 (1.243, 1.628)

Tabela 5.8: Comparação de EQM - Cenário 1: Contribuição do Sill ($\sigma^2_{sill} = 10$).

	EQM (Erro Quadrático Médio)	
Parâmetro / Tendência	EMV	NN-GLS
$\phi = 1$		
Constante	0.592 (0.454, 0.753)	4.858 (4.161, 5.797)
Linear	0.779 (0.607, 0.978)	4.745 (4.013, 5.642)
Complexa	0.711 (0.580, 0.884)	4.656 (3.967, 5.521)
$\phi = 5$		
Constante	0.497 (0.382, 0.619)	2.048 (1.731, 2.371)
Linear	0.598 (0.470, 0.716)	1.996 (1.751, 2.341)
Complexa	0.592 (0.476, 0.752)	2.053 (1.719, 2.445)
$\phi = 8$		
Constante	0.541 (0.427, 0.661)	1.725 (1.479, 2.038)
Linear	0.542 (0.439, 0.690)	1.761 (1.500, 2.046)
Complexa	0.552 (0.447, 0.685)	1.721 (1.488, 1.985)
$\phi = 10$		
Constante	0.538 (0.443, 0.658)	1.629 (1.388, 1.870)
Linear	0.597 (0.488, 0.721)	1.655 (1.402, 1.918)
Complexa	0.639 (0.513, 0.786)	1.674 (1.418, 1.941)

- Desempenho Geral: em todas as combinações de parâmetros, o EMV apresentou EQMs menores, reduzindo frequentemente o erro pela metade em comparação à NN-GLS. A NN-GLS também exibiu maior variabilidade que o EMV.
- ($\phi = 1$): observou-se uma diminuição na performance da NN-GLS quando a dependência espacial é de curto alcance. No cenário de patamar elevado (Tabela 5.8), o EQM da rede atingiu 4,858, em contraste com o EMV (0,592).
- Estruturas para as médias possuem resultados similares: embora haja diferença nos resultados conforme os valores de tendência. Essas mudanças não impactaram os resultados dos métodos ao ponto de vermos a NN-GLS, que tem resultados melhores conforme a complexidade aumenta, se destacando em médias complexas comparada ao EMV.

Embora o EMV permaneça como o estimador mais preciso em cenários ideais e estacionários, a NN-GLS monstrou sua capacidade de adaptação. Seus maiores valores de EQM são compensados pela flexibilidade de modelagem automática da tendência, apresentando melhoria relativa de desempenho justamente nos cenários de maior complexidade estrutural (ϕ alto e tendências não-lineares), que são frequentemente os mais desafiadores em aplicações reais.

5.2.3 Cenário 2

A segunda simulação é inspirada no artigo Zhan and Datta (2025), nessa simulação foram geradas 500 observações na região $[0,1] \times [0,1]$, com média dependendo da função:

$$\mu(s) = \frac{1}{6}(10\sin(\pi x_1(s)x_2(s)) + 20(x_3(s) - 0.5)^2 + 10x_4(s) + 5x_5(s)),$$

em que $\sigma^2 = (1,5)$, $\phi = (\frac{1}{\sqrt{2}}, \frac{6}{\sqrt{2}})$ e $\tau^2/\sigma^2 = 0.1$, fornecendo 4 combinações de campos gaussianos. Os valores do vetor $\mathbf{x}$ foram gerados de uma $X \sim U[0,1]$.

Para a simulação de Monte Carlo foram consideradas 200 repetições para cada uma dessas combinações.

5.2.4 Comparação dos Resultados - Cenário 2

As Tabelas 5.9 e 5.10 apresentam os resultados da segunda etapa de simulação, em que é mostrado a mediana os quartis da estimação de cada modelo. Após isso as Tabelas 5.11 e 5.12 mostram a mesma informação para o EQM. Essa métrica foi escolhida perante ao grande número de outliers, que ficará vísivel na Figura 5.2

Tabela 5.9: Resumo Comparativo de Estimação (EMV) - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 1$)

	σ^2 (Patamar)		ϕ (Alcance)		τ^2 (Nugget)	
	EMV	NN-GLS	EMV	NN-GLS	EMV	NN-GLS
$\phi = 1/\sqrt{2}$	0.554 (0.408, 0.806)	0.527 (0.389, 0.772)	0.346 (0.230, 0.520)	0.353 (0.214, 0.520)	0.475 (0.453, 0.490)	0.555 (0.400, 0.787)
$\phi = 6/\sqrt{2}$	0.133 (0.074, 0.230)	0.123 (0.066, 0.200)	0.304 (0.144, 0.723)	0.325 (0.128, 0.871)	0.857 (0.812, 0.897)	4.765 (2.942, 9.323)

Tabela 5.10: Resumo Comparativo de Estimação (EMV) - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 5$)

	σ^2 (Patamar)		ϕ (Alcance)		τ^2 (Nugget)	
	EMV	NN-GLS	EMV	NN-GLS	EMV	NN-GLS
$\phi = 1/\sqrt{2}$	2.739 (1.963, 4.103)	2.641 (1.916, 4.485)	0.417 (0.295, 0.642)	0.366 (0.254, 0.616)	0.549 (0.516, 0.579)	0.120 (0.078, 0.172)
$\phi = 6/\sqrt{2}$	0.684 (0.400, 1.031)	0.585 (0.396, 0.852)	0.536 (0.319, 0.943)	0.450 (0.295, 0.722)	0.821 (0.782, 0.860)	1.024 (0.687, 1.550)

Tabela 5.11: Comparação de EQM - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 1$)

	EQM (Erro Quadrático Médio)	
Parâmetro (ϕ)	EMV	NN-GLS
$1/\sqrt{2}$	0.521 (0.475, 0.573)	0.522 (0.443, 0.585)
$6/\sqrt{2}$	0.937 (0.821, 1.065)	1.436 (0.927, 2.979)

Tabela 5.12: Comparação de EQM - Cenário 2: Contribuição do Sill ($\sigma^2_{sill} = 5$)

	EQM (Erro Quadrático Médio)	
Parâmetro (ϕ)	EMV	NN-GLS
$1/\sqrt{2}$	0.424 (0.358, 0.494)	0.704 (0.544, 0.834)
$6/\sqrt{2}$	0.877 (0.783, 0.999)	0.864 (0.737, 1.111)

A análise dos resultados para o Cenário 2 revela padrões quanto à capacidade dos modelos em lidar com a complexidade estrutural simulada. Abaixo, destacam-se as principais observações referentes à estimação dos parâmetros de covariância e ao desempenho preditivo:

5.2.5 Estimação dos Parâmetros

- σ^2: tanto o EMV quanto a NN-GLS subestimaram o patamar. No caso de $\sigma^2 = 5$, as estimativas de ambos os métodos ficaram aproximadamente entre 2 e 4. Isso sugere que parte da variabilidade estrutural foi interpretada como tendência ou diluída.

- Efeito Pepita τ^2: a rede neural demonstrou comportamento de superestimar ou subestimar τ^2. Com $\sigma^2 = 1$ e longo alcance, a NN-GLS estimou um efeito pepita alto ($\approx 4,76$), indicando incapacidade de discernir a estrutura espacial fraca. E no cenário com ($\sigma^2 = 5$) e curto alcance, a rede subestimou o pepita ($\approx 0,12$).

- O tempo de estimação dos parâmetros com o EMV foi de 3,43 segundos, enquanto o da NN-GLS foi de 1,13. Isso mostra que mesmo para uma quantidade de pontos consideravelmente pequena o NN-GLS foi 3 vezes mais rápido.

5.2.6 Desempenho (EQM)

- Curto Alcance ($\phi = 1/\sqrt{2}$): o EMV manteve a superioridade, apresentando menor erro tanto para baixo quanto para alto patamar.

- Longo Alcance ($\phi = 6/\sqrt{2}$): na Tabela 5.12 (Sill=5), a NN-GLS apresentou um EQM inferior ao EMV (0,864 contra 0,877). Isso sugere que, na presença de um sinal espacial forte e de longo alcance combinado com complexidade estrutural, a flexibilidade da NN-GLS em modelar a tendência não-linear compensa a sua ineficiência na estimação dos parâmetros. Mesmo assim, a variabilidade dela foi maior que a do EMV.

Para entendermos melhor a distribuição dos parâmetros estimados e do EQM foram feitos gráficos boxplot com o pacote ggplot2 Wickham (2016) como visto na Figura 5.2, a linha tracejada em vermelho indica o verdadeiro valor do parâmetro.

Os gráficos da NN-GLS apresentam, em geral, maior amplitude interquartílica, indicando menor precisão nas estimativas pontuais. Destaca-se o comportamento errático na estimação do efeito pepita (τ^2) para o cenário de longo alcance, onde a distribuição da rede neural colapsa para valores extremos ou apresenta grande dispersão, enquanto o EMV mantém-se estável. Devido à grande quantidade de outliers, os limites dos gráficos foram fixados em (20, 15, 10, 10).

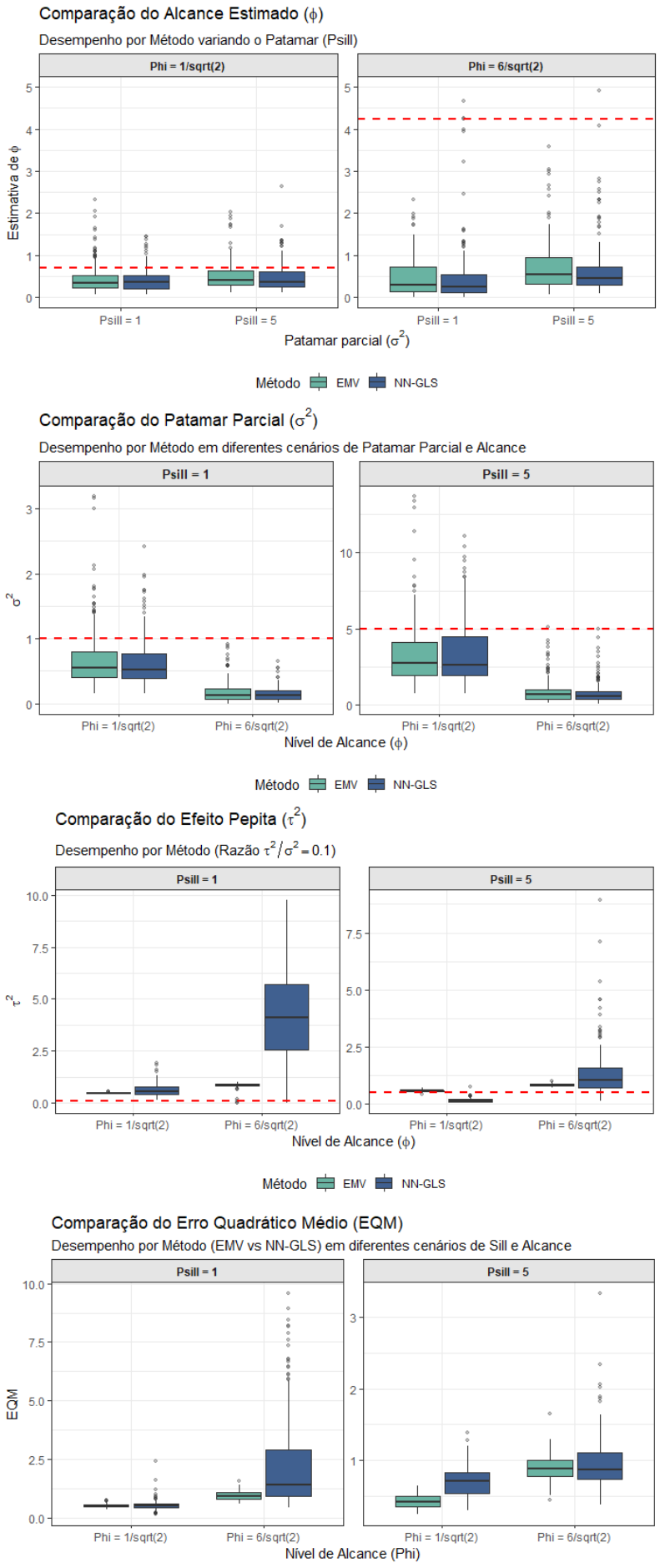

Figura 5.2: Gráficos de comparação dos valores estimados dos parâmetros e do EQM.

6 Dados Reais

Este estudo utiliza dados de estações meteorológicas automáticas espalhadas pelo território brasileiro disponíveis em Instituto Nacional de Meteorologia (INMET) (2025). O período selecionado foi de 1 ano, de 01/05/2024 até 01/05/2025.

Foi necessário realizar uma agregação temporal das observações originais. Para as covariáveis explicativas, calculou-se a média aritmética anual dos registros, capturando o comportamento central de cada variável no período. Já para a variável resposta (precipitação), optou-se pelo somatório do período, representando o volume total de chuvas. As variáveis resultantes foram definidas como:

- Precipitação - Nossa variável resposta e o somatório de precipitação em milímetros observada no período,
- Pressão - Média da pressão atmosférica em hectopascals (hPA) no período,
- Temperatura - Média da temperatura em graus no período,
- Umidade - Média da umidade em porcentagem no período.

O conjunto de dados também inclui a latitude e a longitude de cada variável que define a posição de cada estação no mapa e serve para a criação das arestas dos grafos. Na rede, todas as covariáveis, inclusive as coordenadas, são individualmente normalizadas.

6.1 Arquitetura da rede

A rede neural artificial utiliza o método NN-GLS foi estruturada como um MLP sequencial, implementado via biblioteca PyTorch. A arquitetura é composta por uma camada de entrada ajustada à dimensão das covariáveis (p), seguida por quatro camadas ocultas contendo, respectivamente, 64, 32, 16 e 8 neurônios. Para introduzir não-linearidade ao modelo, aplicou-se a função de ativação ReLU (Rectified Linear Unit) após cada camada oculta. O processo de treinamento foi configurado com uma taxa de aprendizado de 0,01 e um critério de parada antecipada de 0,001 para controle de convergência. A divisão do banco foi mudada para 45% de teste, 10% de validação e 45% de treinamento, mesma divisão utilizada pelos autores Zhan and Datta (2025).

6.2 Resultados

Na Figura 6.1 são apresentadas as predições da precipitação estimadas pela krigagem usando o EMV e a NN-GLS. Para a confecção do gráfico, foi utilizado o pacote leaflet Cheng et al. (2025) do R. Ao analisar os valores extremos de precipitação, nota-se uma distinção importante entre as abordagens. Enquanto o método EMV preservou a variabilidade local, capturando picos de precipitação com maior intensidade (áreas mais claras), o modelo NN-GLS apresentou um comportamento de suavização dessas anomalias. Embora gere um campo visualmente contínuo, essa suavização tende a subestimar a magnitude dos extremos, atenuando as observações de alta precipitação em direção à média local.

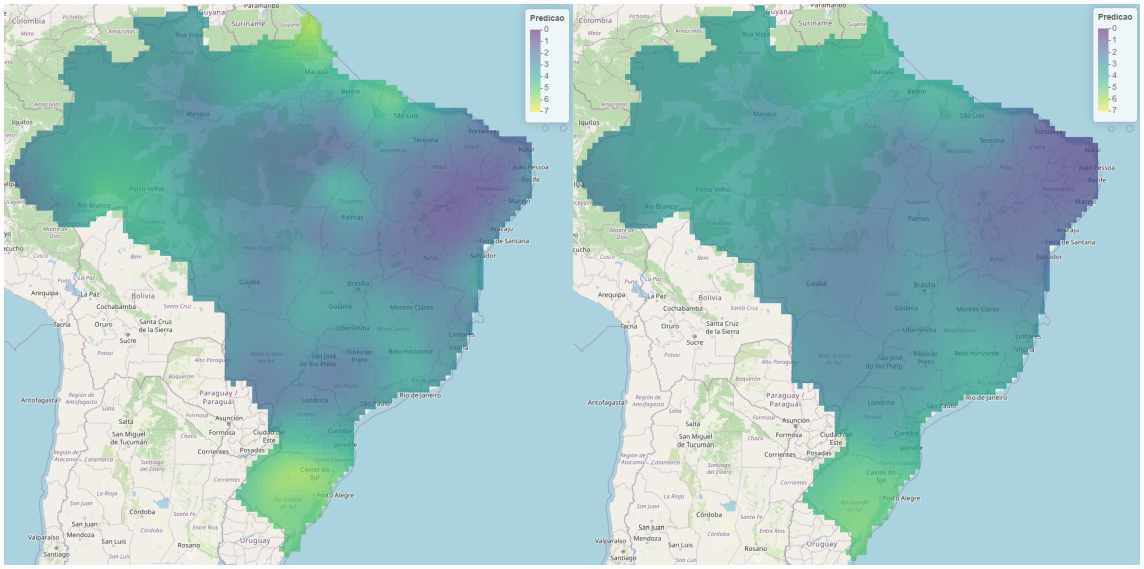

Figura 6.1: Mapa da predição da quantidade total de precipitação no período de 01/05/2024 até 01/05/2025. na esquerda EMV e na direita NN-GLS.

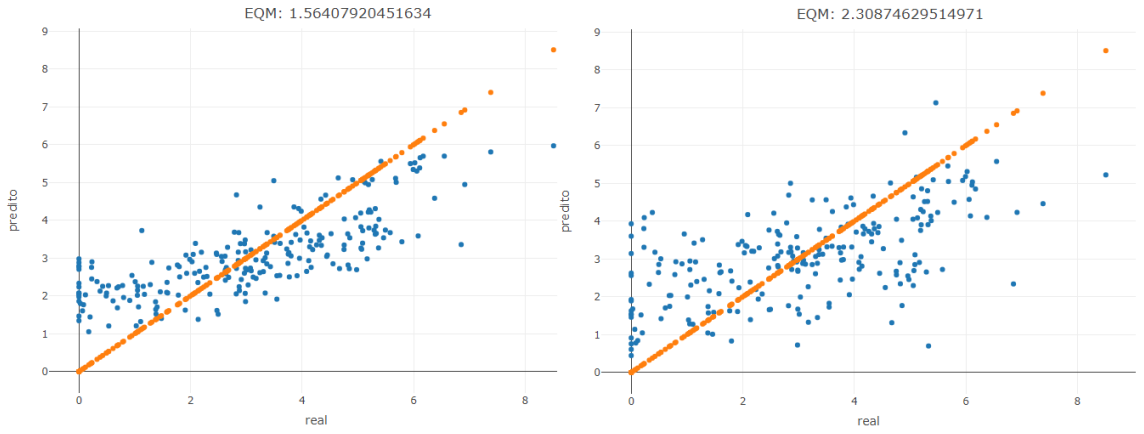

Figura 6.2: Gráfico do EQM da krigagem na esquerda EMV e na direita NN-GLS.

A aplicação dos métodos aos dados de precipitação acumulada (01/05/2024 a 01/05/2025) revelou diferenças marcantes na capacidade de cada modelo em capturar a variabilidade espacial do fenômeno.

- EQM: o EMV apresentou um EQM de 1,56, menor que o observado para a NN-GLS (2,31). Isso é corroborado visualmente pelos gráficos de dispersão na Figura 6.2, onde os pontos da Krigagem (esquerda) estão mais concentrados em torno da reta identidade ($y = x$), enquanto a NN-GLS (direita) exibe maior dispersão, indicando menor precisão e maior variância nos resíduos.

- Parâmetros Estimados: a diferença no desempenho pode ser explicada pela diferença na estimação dos parâmetros da função de covariância. O Alcance do EMV ($\hat{\phi} = 15.68$) é maior do que do NN-GLS ($\hat{\phi} = 8.83$). Isso indica que a Krigagem conseguiu capturar uma dependência espacial de longa escala que a rede neural ignorou. Ao restringir o alcance, a NN-GLS limitou a influência de vizinhos distantes, perdendo informação crucial para a predição em um país de dimensões continentais como o Brasil. Na estimação do patamar (σ^2): A NN-GLS subestimou a variância estrutural ($\hat{\sigma^2} = 0.94$) em comparação ao EMV ($\hat{\sigma^2} = 1.47$). Ao estimar uma variância menor, a rede interpretou parte da variação espacial real como se fosse menor do que realmente é.

- Predição: a NN-GLS aparenta ser mais suavizada que a da krigagem (Figura 6.1) é uma consequência direta da subestimação dos parâmetros discutida acima. Como a NN-GLS estimou um patamar menor e um alcance mais curto, ela produziu uma superfície de predição tendendo à média local e falhando em alcançar os picos extremos de precipitação. A Krigagem com EMV, com parâmetros que descrevem uma estrutura espacial mais forte e abrangente, gerou um mapa com contrastes mais definidos, refletindo melhor a natureza irregular da chuva e resultando, consequentemente, em um menor erro de predição.

- Pressão: apresentou um coeficiente $\hat{\beta}_1 \approx -0.078$. Isso indica uma relação inversa com a variável resposta: valores mais altos desta covariável tendem a diminuir a precipitação esperada.

- Temperatura: exibiu um coeficiente positivo de magnitude similar $\hat{\beta}_2 \approx 0.077$, sugerindo que o aumento desta variável contribui positivamente para a ocorrência de precipitação.

- Umidade: apresentou um coeficiente próximo de zero $\hat{\beta}_3 \approx 0.003$. Este valor indica que, na presença das outras duas variáveis, a contribuição desta covariável para estimar a tendência média foi pequena.

Ao comparar os coeficientes globais com os gráficos de dependência parcial da NN-GLS (Figura 6.3), observa-se uma concordância na direção das tendências para pressão e temperatura. Contudo, o EMV falha em capturar a complexidade da Umidade. Enquanto o coeficiente $\hat{\beta}_3 \approx 0$ sugere irrelevância, o PDP revela um comportamento não-linear que o modelo de regressão simples não conseguiu detectar.

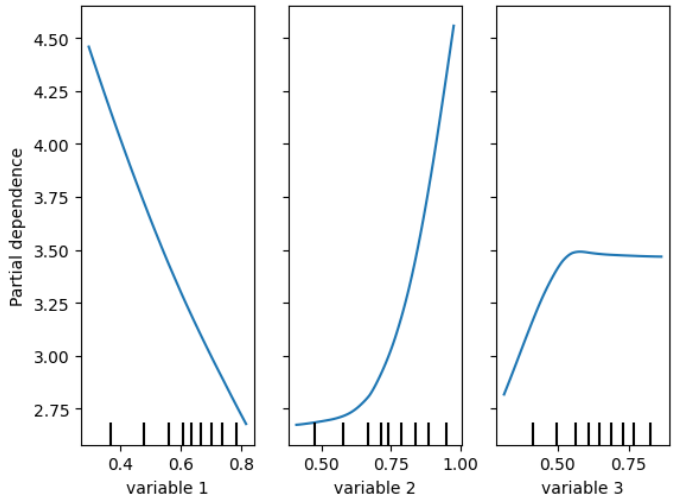

Figura 6.3: Gráficos de dependências parciais da NN-GLS em ordem, de esquerda para direita, das covariáveis: pressão, temperatura e umidade.

7 Conclusão

Este trabalho aplicou o método de rede neural NN-GLS em comparação ao EMV em dados climáticos e sintéticos. Os resultados mostraram que a NN-GLS não se mostrou mais eficaz que o EMV em muitos casos. No entanto, foi observada, nos resultados das simulações, uma melhora nos resultados da NN-GLS conforme o aumento da complexidade da função da média do processo gaussiano. Adicionalmente, observou-se que a NN-GLS tende a subestimar o efeito pepita, incorporando parte da variabilidade não correlacionada à estrutura da tendência.

Outra direção para trabalhos futuros é a incorporação da dependência temporal. Como os fenômenos climáticos são intrinsecamente dinâmicos, a extensão da arquitetura NN-GLS para incluir a dimensão temporal permitiria capturar a evolução dos padrões de precipitação e a sazonalidade, superando as limitações de uma análise puramente espacial estática.

Além disso, para estudos futuros, sugere-se que a NN-GLS leve em conta não apenas os vizinhos mais próximos, mas também pontos mais distantes. Casos como teleconexões atmosféricas, como o El Niño Ashok et al. (2007), efeito buraco Pyrcz and Deutsch (2003) ou efeito de blindagem Stein (2002), são fenômenos climáticos que possuem essa estrutura de longa dependencia que pode não ser captada pelo modelo NN-GLS.

Por fim, as simulações foram realizadas somente com um campo aleatório gaussiano e com funções de média que supomos serem complexas. Recomenda-se a simulação de uma grade com dois campos gaussianos com médias diferentes para garantir a violação dos pressupostos do EMV; nesse caso, a rede neural deve ter suas vantagens completamente exploradas.

Referências Bibliográficas

S. Akansha. Over-squashing in graph neural networks: A comprehensive survey, 2025. URL https://arxiv.org/abs/2308.15568.

K. Ashok, S. K. Behera, S. A. Rao, H. Weng, and T. Yamagata. El niño modoki and its possible teleconnection. *Journal of Geophysical Research: Oceans*, 112(C11), 2007.

K. Ashrafi, M. Shafiepour, L. Ghasemi, and B. Araabi. Prediction of climate change induced temperature rise in regional scale using neural network. *International Journal of Environmental Research*, 6(3):677–688, 2012.

S. Atanlogun, O. Edwin, and Y. Afolabi. On comparative modeling of gls and ols estimating techniques. *International Journal of Scientific & Technology Research*, 3(1):125–128, 2014.

G. Bohling. Introduction to geostatistics and variogram analysis. *Kansas geological survey*, 1(10):1–20, 2005.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

H. Cai, V. W. Zheng, and K. C.-C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018.

J. Cheng, B. Schloerke, B. Karambelkar, Y. Xie, and G. Aden-Buie. *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*, 2025. URL https://rstudio.github.io/leaflet/. R package version 2.2.3.9000.

N. Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.

A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand. Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016.

R. Diestel. *Graph theory*. Springer Nature, 2025.

P. Diggle, P. Ribeiro, and M.-b. Geostatistics. Springer series in statistics. *Springer. Djelouah K., Frasheri D., Valentini F., D'Onghia AM and Digiaro M.(2014). Direct tissue blot immunoassay for detection of Xylella fastidiosa in olive trees. Phytopathologia Mediterranea*, 53(3):559–564, 2007.

P. J. Diggle, J. A. Tawn, and R. A. Moyeed. Model-based geostatistics. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 47(3):299–350, 1998.

P. J. Diggle, P. J. Ribeiro Jr, and O. F. Christensen. An introduction to model-based geostatistics. In *Spatial statistics and computational methods*, pages 43–86. Springer, 2003.

P. Goovaerts. *Geostatistics for natural resources evaluation*. Oxford university press, 1997.

K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu. Vision gnn: An image is worth graph of nodes. *Advances in neural information processing systems*, 35:8291–8303, 2022.

Instituto Nacional de Meteorologia (INMET). Tabela de estações - inmet. https://tempo.inmet.gov.br/TabelaEstacoes/A001, 2025. Acessado em: 21 de maio de 2025.

J. Jiang, Z.-G. Huang, C. Grebogi, and Y.-C. Lai. Predicting extreme events from data using deep machine learning: When and where. *Phys. Rev. Res.*, 4:023028, Apr 2022. doi: 10.1103/PhysRevResearch.4.023028. URL https://link.aps.org/doi/10.1103/PhysRevResearch.4.023028.

W. Jiang and J. Luo. Graph neural network for traffic forecasting: A survey. *Expert systems with applications*, 207:117921, 2022.

M. Kabir, U. E. Habiba, W. Khan, A. Shah, S. Rahim, P. R. D. los Rios-Escalante, Z.-U.-R. Farooqi, L. Ali, and M. Shafiq. Climate change due to increasing concentration of carbon dioxide and its impacts on environment in 21st century; a mini review. *Journal of King Saud University - Science*, 35(5):102693, 2023. ISSN 1018-3647. doi: https://doi.org/10.1016/j.jksus.2023.102693. URL https://www.sciencedirect.com/science/article/pii/S1018364723001556.

S. Kim, H. Kim, J. Lee, S. Yoon, S. E. Kahou, K. Kashinath, and M. Prabhat. Deep-hurricane-tracker: Tracking and forecasting extreme climate events. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1761–1769, 2019. doi: 10.1109/WACV.2019.00192.

R. Knutti, T. Stocker, F. Joos, and G.-K. Plattner. Probabilistic climate change projections using neural networks. *Climate Dynamics*, 21:257–272, 2003.

D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

M. Lanfredi, R. Coppola, M. D'Emilio, V. Imbrenda, M. Macchiato, and T. Simoniello. A geostatistics-assisted approach to the deterministic approximation of climate data. *Environmental Modelling & Software*, 66:69–77, 2015.

W.-D. Li, K. Hu, C. Larsen, Y. Wu, S. Alford, C. Woo, S. M. Dunn, H. Tang, M. Naim, D. Nguyen, et al. Combining induction and transduction for abstract reasoning. *arXiv preprint arXiv:2411.02272*, 2024.

X. Li, L. Sun, M. Ling, and Y. Peng. A survey of graph neural network based recommendation in social networks. *Neurocomputing*, 549:126441, 2023.

J. A. Lin, S. Ament, M. Balandat, D. Eriksson, J. M. Hernández-Lobato, and E. Bakshy. Scalable gaussian processes with latent kronecker structure, 2025. URL https://arxiv.org/abs/2506.06895.

D. Maraun, F. Wetterhall, A. M. Ireson, R. E. Chandler, E. J. Kendon, M. Widmann, S. Brienen, H. W. Rust, T. Sauter, M. Themeßl, et al. Precipitation downscaling under climate change: Recent developments to bridge the gap between dynamical models and the end user. *Reviews of geophysics*, 48(3), 2010.

M. Marinescu. Explaining and connecting kriging with gaussian process regression. *arXiv preprint arXiv:2408.02331*, 2024.

G. Matheron. Principles of geostatistics. *Econ. Geol.*, 58(8):1246–1266, Dec. 1963.

P. Moraga. *Geospatial health data: Modeling and visualization with R-INLA and shiny*. Chapman and Hall/CRC, 2019.

E. National Academies of Sciences and Medicine. *Attribution of Extreme Weather Events in the Context of Climate Change*. The National Academies Press, Washington, DC, 2016. ISBN 978-0-309-38094-2. doi: 10.17226/21852. URL https://nap.nationalacademies.org/catalog/21852/attribution-of-extreme-weather-events-in-the-context-of-climate-change.

M. Pyrcz and C. Deutsch. The whole story on the hole effect. *Geostatistical Association of Australasia, Newsletter*, 18:3–5, 2003.

C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

P. J. Ribeiro Jr and P. J. Diggle. *geoR: Analysis of Geostatistical Data*, 2025. URL https://CRAN.R-project.org/package=geoR. R package version 1.9-6.

T. K. Rusch, M. M. Bronstein, and S. Mishra. A survey on oversmoothing in graph neural networks, 2023. URL https://arxiv.org/abs/2303.10993.

B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 6(8), Aug. 2021a.

B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021b. doi: 10.23915/distill.00033. https://distill.pub/2021/gnn-intro.

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *Trans. Neur. Netw.*, 20(1):61–80, Jan. 2009. ISSN 1045-9227. doi: 10.1109/TNN.2008.2005605. URL https://doi.org/10.1109/TNN.2008.2005605.

M. Sherman. *Spatial statistics and spatio-temporal data: covariance functions and directional properties*. John Wiley & Sons, 2011.

M. L. Stein. The screening effect in kriging. *The Annals of Statistics*, 30(1):298–323, 2002.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

von Merian-Erben. pontes. https://www.preussenchronik.de/bild_jsp/key=bild_kathe2.html, 1652. Acessado em: 08/12/25).

D. Wang, Y. Chen, M. Jarin, and X. Xie. Increasingly frequent extreme weather events urge the development of point-of-use water treatment systems. *npj Clean Water*, 5(1):36, Aug 2022. ISSN 2059-7037. doi: 10.1038/s41545-022-00182-1. URL https://doi.org/10.1038/s41545-022-00182-1.

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, and T. Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

E. Williamson. Lists, decisions and graphs. *S. Gill Williamson*, page 12, 2010.

W. Zhan and A. Datta. Neural networks for geospatial data. *Journal of the American Statistical Association*, 120(549):535–547, 2025.

P. Zhang and G. Chartrand. *Introduction to graph theory*, volume 2. Tata McGraw-Hill New York, 2006.