

1. Introdução

1.1. Contextualização

Um sistema linear do tipo $Ax = b$ representa um conjunto de equações lineares simultâneas, q solução desse sistema consiste em determinar os valores de x que satisfaçam todas as equações simultaneamente (Trefethen & Bau, 1997; Golub & Van Loan, 2013; Saad, 2003). Esses sistemas são amplamente aplicados em diversas áreas da ciência e engenharia, como circuitos elétricos, dinâmica de fluidos, análise estrutural e física computacional, devido à sua capacidade de modelar fenômenos complexos (Saad, 2003).

Para resolvê-los, existem métodos diretos, como a eliminação de Gauss e as decomposições LU e QR, que transformam o sistema em uma forma mais simples para obter a solução exata (Golub & Van Loan, 2013). Contudo, esses métodos tornam-se inviáveis para sistemas de grande porte ou matrizes esparsas, caracterizadas pela presença de muitos elementos zero, devido ao elevado custo computacional e de memória (Saad, 2003).

Como alternativa, os métodos iterativos iniciam com uma estimativa inicial para x e vão ajustando de forma progressiva até se aproximar da solução correta, sendo particularmente eficazes em sistemas grandes e esparsos, comuns em aplicações reais (Saad, 2003). Exemplos incluem o método de Jacobi, o Gauss-Seidel e o Gradiente Conjugado, que oferecem soluções com menor custo computacional (Saad, 2003; Trefethen & Bau, 1997).

A seleção do método adequado depende de características da matriz A, como a quantidade de zeros, simetria, sensibilidade a erros (medida pelo número de condição) e dominância diagonal, evidenciando a necessidade de abordagens adaptáveis para lidar com problemas diversos (Saad, 2003; Golub & Van Loan, 2013; Trefethen & Bau, 1997).

1.2. Identificação do Problema

A escolha inadequada de métodos iterativos para resolver sistemas lineares representa um desafio significativo, especialmente em aplicações reais com matrizes complexas e de grande escala. A seleção manual, frequentemente baseada em heurísticas empíricas, pode levar a soluções subótimas, resultando em ineficiência no tempo de processamento e no uso de recursos computacionais (Saad, 2003).

Essa dependência de processos manuais exige conhecimento especializado, nem sempre disponível, e configura um problema central, pois a identificação do método mais adequado é dificultada pela variabilidade das características das matrizes, como esparsidade, simetria e número de condição (Golub; Van Loan, 2013; Trefethen; Bau, 1997).

1.3. Justificativa

O desenvolvimento de uma abordagem preditiva baseada em aprendizado de máquina oferece um avanço promissor para superar as limitações das heurísticas. Essa abordagem permite analisar características matriciais, como esparsidade e dominância diagonal, para recomendar o método iterativo mais eficiente, reduzindo a dependência de escolhas empíricas e otimizando o desempenho computacional em simulações críticas

(Saad, 2003). Dessa forma, justifica-se a necessidade de um sistema que automatize e aprimore a decisão.

1.4. Objetivos

1.4.1. Objetivo Geral

Desenvolver um sistema computacional que utilize aprendizado de máquina para recomendar métodos iterativos adequados à resolução de sistemas lineares, com base nas características estruturais das matrizes.

1.4.2. Objetivos Específicos

- i. Estudar e implementar métodos iterativos, como Jacobi, Gauss-Seidel e Gradiente Conjugado.
- ii. Gerar bases de dados com sistemas lineares variados e extrair métricas relevantes das matrizes.
- iii. Treinar modelos de aprendizado supervisionado para predizer o método iterativo mais eficiente.
- iv. Avaliar o desempenho dos modelos em dados sintéticos e reais.

1.5. Contribuições

Espera-se que este trabalho contribua com a criação de um sistema funcional que auxilie na escolha otimizada de métodos iterativos, reduzindo o esforço manual e melhorando a eficiência computacional. A base de dados gerada e o modelo treinado poderão servir como ferramentas para futuras pesquisas, oferecendo uma abordagem sistemática e adaptável a diferentes problemas lineares.

2. Referencial Teórico

Este capítulo apresenta os conceitos fundamentais relacionados aos sistemas lineares, métodos de resolução e características das matrizes.

2.1. Sistemas Lineares

Sistemas lineares da forma $Ax = b$, onde A é uma matriz $m \times n$ de coeficientes, x é um vetor coluna $n \times 1$ de variáveis incógnitas e b é um vetor coluna $m \times 1$ de termos constantes, a solução de tais sistemas envolve determinar os valores de x que satisfazem simultaneamente todas as equações (Trefethen; Bau, 1997; Golub; Van Loan, 2013; Saad, 2003).

2.2. Métodos Iterativos

Métodos iterativos são bem úteis para resolver sistemas lineares esparsos, onde a matriz A possui muitos elementos zero, comuns em aplicações como simulações numéricas em engenharia e física computacional. Esses métodos iniciam com uma aproximação inicial

$x^{(0)}$ e refinam iterativamente até convergir para a solução exata, explorando a estrutura esparsa para minimizar operações computacionais (Saad, 2003). Diferentemente dos métodos diretos, que podem ser ineficientes para matrizes grandes devido ao custo de $O(n^3)$, os métodos iterativos têm custo por iteração proporcional ao número de elementos não nulos (Golub; Van Loan, 2013).

2.2.1. Método de Jacobi

O método de Jacobi decompõe a matriz $A = D + L + U$, onde D é a diagonal, L é a triangular inferior estrita e U é a triangular superior estrita. Iniciando com uma aproximação inicial $x^{(0)}$, para $k = 0, 1, 2 \dots$ até a convergência:

$$x^{(k+1)} = D^{-1} \cdot (b - (L + U) \cdot x^{(k)})$$

ou, componente a componente:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \cdot (b_i - \sum_{j \neq i} a_{ij} \cdot x_j^{(k)}), \quad i = 1, \dots, n$$

Cada componente é atualizado independentemente, facilitando o paralelismo (Saad, 2003). A convergência é garantida para matrizes estritamente diagonalmente dominantes, com raio espectral do operador de iteração menor que 1, mas pode ser lento em sistemas mal condicionados (Trefethen; Bau, 1997).

2.2.2. Método de Gauss-Seidel

Similar ao Jacobi, mas atualiza componentes sequencialmente, incorporando valores novos imediatamente. Iniciando com uma aproximação inicial $x^{(0)}$, para $k = 0, 1, 2 \dots$ até a convergência:

$$x^{(k+1)} = (D + L)^{-1} \cdot (b - U \cdot x^{(k)})$$

ou, componente a componente:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \cdot (b_i - \sum_{j < i} a_{ij} \cdot x_j^{(k+1)} - \sum_{j > i} a_{ij} \cdot x_j^{(k)}), \quad i = 1, \dots, n$$

Isso melhora a convergência em comparação ao Jacobi para matrizes positivas definidas. Para matrizes simétricas positivas definidas (SPD), converge para qualquer inicialização (Golub; Van Loan, 2013, p. 616). Requer processamento sequencial, limitando o paralelismo, mas é mais eficiente em termos de iterações em relação ao método anterior.

2.2.3. Método do Gradiente Conjugado

Para matrizes simétricas positivas definidas, o método do Gradiente Conjugado minimiza a função quadrática $f(x) = \frac{1}{2} \cdot x^T \cdot A \cdot x - b^T \cdot x$, gerando direções conjugadas ortogonais em relação ao produto interno induzido por A (Trefethen; Bau, 1997). Iniciando com x_0 , computando $r_0 = b - A \cdot x_0$, $p_0 = r_0$, para $k = 1, 2 \dots$ até a convergência:

$$r^{(k)} = b - A \cdot x^{(k)}, p^{(k)} = r^{(k)} + \beta \cdot p^{(k-1)}$$

Com convergência em no máximo n passos, com taxa de convergência dependente do número de condição $\kappa(A)$.

2.3. Características das Matrizes

A eficiência dos métodos iterativos depende de certas propriedades matriciais. Dentre elas, temos a esparsidade medida por $1 - \frac{\text{número de elementos não zero}}{\text{tamanho total}}$ (Golub; Van Loan, 2013), a simetria, verificada por $A = A^T$ e que facilita as decomposições ortogonais (Trefethen; Bau, 1997), o número de condição $\kappa(A)$ indica sensibilidade a perturbações; valores altos demandam pré-condicionadores para acelerar convergência (Saad, 2003) e a dominância diagonal, onde $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$, que garante a convergência em métodos como Jacobi e Gauss-Seidel (Golub; Van Loan, 2013).

3. Metodologia

Neste capítulo serão abordados todos os procedimentos metodológicos e ferramentas utilizadas para o desenvolvimento do sistema computacional proposto.

A abordagem adota uma análise quantitativa, utilizando estatísticas para avaliar o desempenho preditivo do modelo, com fundamentos teóricos baseados na álgebra linear e em aprendizado de máquina.

A natureza dos objetivos é predominantemente aplicada, voltada para a criação de uma ferramenta prática que auxilie na resolução de problemas reais, e exploratória, ao buscar identificar padrões e relações entre características matriciais e o desempenho de métodos iterativos por meio do aprendizado de máquina.

3.1. Implementação dos Métodos Iterativos

Os métodos iterativos (Jacobi, Gauss-Seidel e Gradiente Conjugado) foram implementados em Python, utilizando a biblioteca NumPy para cálculos matriciais. A implementação seguiu algoritmos descritos por Golub & Van Loan (2013, p. 611-639) e Saad (2003, p. 105-114, p. 196-203). Cada método foi testado inicialmente em matrizes sintéticas com diferentes tamanhos para verificar convergência, usando tolerância de 10^{-6} e limite de 1000 (mil) iterações, com validação baseada no erro relativo $\frac{\|Ax-b\|}{\|b\|}$.

3.2. Coleta de Dados

Os dados utilizados neste trabalho compreende dois conjuntos principais: dados sintéticos e dados reais.

Os dados sintéticos foram gerados para incluir sistemas lineares variados, com matrizes de diferentes tamanhos e propriedades (esparsidade, simetria, dominância diagonal). Exemplos iniciais foram gerados aleatoriamente conforme as métricas utilizadas, como matrizes tridiagonais e matrizes diagonalmente dominantes, para validação preliminar dos métodos iterativos.

Os dados reais ainda não foram implementados.

3.3. Geração da Base de Dados

Uma base de dados foi construída a partir dos sistemas lineares coletados, incluindo tanto dados sintéticos quanto reais. Para cada matriz A , extraiu-se um conjunto de métricas relevantes: esparsidade, simetria, número de condição e dominância diagonal. As métricas foram calculadas usando funções de álgebra linear e armazenadas em um arquivo .csv para posterior análise.

3.4. Treinamento do Modelo de Aprendizado Supervisionado

Ainda não implementado, o modelo preditivo será treinado utilizando as bibliotecas Scikit-learn e PyTorch. As métricas extraídas servirão como features, e o rótulo de melhor método iterativo ainda será discutido.

3.5. Avaliação do Desempenho

O desempenho dos modelos será avaliado em dados sintéticos e reais, utilizando métricas quantitativas como convergência, número de iterações, tempo de execução e resíduo final. Para cada sistema, o método predito será comparado com o método ideal (determinado experimentalmente), calculando a taxa de acerto e o erro médio.

3.6. Justificativa da Metodologia

A abordagem quantitativa foi escolhida para permitir uma análise estatística objetiva do desempenho preditivo, alinhada aos objetivos do trabalho. O uso de Python e bibliotecas como NumPy, Scikit-learn e PyTorch foi baseado na acessibilidade e na robustez dessas ferramentas, amplamente utilizadas em pesquisa computacional.

3.7. Ferramentas Utilizadas

O ambiente de desenvolvimento foi baseado em Python 3.11, escolhido por sua flexibilidade e suporte a bibliotecas especializadas.

A biblioteca NumPy foi utilizada para as operações matriciais eficientes, além da geração de dados sintéticos usando funções aleatórias.

Para o treinamento do modelo preditivo, será utilizado bibliotecas como Scikit-learn e PyTorch.

4. Cronograma

Etapas	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6
Revisão bibliográfica	X	X				
Implementação dos métodos iterativos		X	X			
Geração de base de dados e extração de métricas matriciais			X	X		
Treinamento do modelo de aprendizado supervisionado				X	X	
Avaliação de desempenho em dados sintéticos e reais					X	X
Escrita da Monografia	X	X	X	X	X	X