Relatório de CPD

IMPLEMENTAÇÃO DA BIBLIOTECA MPI

Nome: Enoque Rogério
Número: 20201649

Nome: Derby Cândido
Número: 20200074

Nome: Helton Hevambi
Número: 20200068

***Abstract*—This paper presents an implementation of a matrix factorization algorithm using the Message Passing Interface (MPI) for parallel computation in recommendation systems. The algorithm decomposes a sparse user-item interaction matrix into two lower-dimensional matrices, enabling prediction of missing values and personalized recommendations. By initializing the matrices with random values and iteratively updating them via gradient descent to minimize the error, the workload is efficiently distributed across multiple processors using MPI. This parallel approach enhances performance by reducing computation time and maintaining prediction accuracy, making it viable for large-scale recommendation tasks.**

Keywords—Matrix Factorization, MPI, Parallel Computing, Recommendation Systems, Gradient Descent, Sparse Matrix, Latent Features, Distributed Computing, High-Performance Computing, Collaborative Filtering.

Introdução

Este relatório descreve a implementação e execução de um algoritmo de factoração de matrizes utilizando o paradigma de programação paralela com a biblioteca MPI (Message Passing Interface). O código em questão tem como objetivo realizar a factoração de uma matriz esparsa, utilizando um modelo de aprendizado de máquina baseado em álgebra linear, para recomendar itens a usuários

Equations

As principais equações utilizadas no código de fatoração de matrizes com MPI, que implementa uma forma de Fatoração de Matriz com Gradiente Descendente para Sistemas de Recomendação

O objetivo é fatorar uma matriz original A4 em duas matrizes menores L e R de tal forma que $A \approx L \times R A \approx L \times R$

Onde:

- A é a matriz original de dimensão $nU \times nI$nU×nI.
- L é a matriz de características dos usuários de dimensão $nU \times nF$nU×nF
- R é a matriz de características dos itens de dimensão $nF \times nI$nF×nI.

Função de Custo

A função de custo que o algoritmo tenta minimizar é baseada no erro quadrático entre os valores conhecidos da matriz original AA e os valores preditos pela multiplicação $L \times R L \times R$:

$J(L,R)=\sum(i,j)\in\Omega(Aij-(L\times R)ij)2$

onde Ω é o conjunto dos índices $(i,j)(i,j)$ para os quais $AijAij$ é conhecido (elementos não nulos).

Erros Comuns

A variação de resultados dependendo do número de processadores usados alterando assim a sua saída como foi o problema encontrado durante o desenvolvimento deste trabalho bem como a dificuldade na partilha de informações entre os processadores ,foram os problemas

LUANDA, 2024