

# Sistemas de Recomendação

Lorenzo Carneiro Magalhães - 2021031505

## 1 Introdução

Os sistemas de recomendação são ferramentas essenciais em muitas aplicações modernas, desde recomendar filmes em plataformas de streaming até sugerir produtos em lojas online. O objetivo principal é prever a preferência ou classificação que um usuário daria a um item, com base em dados anteriores. Neste trabalho, focamos em prever ratings baseados em avaliações anteriores de usuários e itens.

## 2 Estratégias

### 2.1 Pré-processamento dos Dados

Os dados foram lidos a partir de arquivos CSV. As colunas 'UserId:ItemId' foram divididas em duas colunas separadas, 'User' e 'Item', para facilitar o mapeamento e a indexação. Isso foi feito com o auxílio da biblioteca pandas.

### 2.2 Mapeamento de Usuários e Itens

Os usuários e itens únicos foram mapeados para índices numéricos, facilitando o acesso e a manipulação dos dados.

### 2.3 Implementação do funkSVD com Viés

O algoritmo funkSVD foi utilizado para decompor a matriz de ratings em duas matrizes menores, 'P' (usuários) e 'Q' (itens). Além disso, foram introduzidos vieses para usuários e itens, representados pelas matrizes 'bu' e 'bi', respectivamente. O viés global, representado por  $\mu$ , é a média de todos os ratings. A predição do rating é dada por:

$$\hat{r}_{ui} = \mu + b_u + b_i + P_u \cdot Q_i^T$$

Estas matrizes são inicializadas com pequenos valores aleatórios que seguem uma distribuição normal.

#### 2.3.1 Gradiente Descendente em Mini-batches

O algoritmo foi modificado para usar mini-batches. Cada mini-batch é usado para calcular o gradiente e atualizar os pesos. Isso pode acelerar a convergência e é especialmente útil para conjuntos de dados grandes.

### 2.3.2 Taxa de Aprendizado e Regularização

A taxa de aprendizado controla o tamanho do passo em cada atualização dos pesos, enquanto a regularização previne o overfitting. Ambos os parâmetros foram ajustados manualmente, mas um ajuste mais refinado pode melhorar o desempenho. Além disso, a taxa de aprendizado e a regularização são reduzidas multiplicativamente a cada época.

### 2.3.3 Épocas

O algoritmo foi executado por um número fixo de épocas. Em cada época, os ratings são embaralhados para garantir que o modelo não se ajuste a uma ordem específica.

## 2.4 Predição

Uma função de predição foi definida para prever o rating de um usuário para um item específico. Esta função utiliza as matrizes ‘P’, ‘Q’, ‘bu’, e ‘bi’, além do viés global  $\mu$ , para calcular a predição.

## 2.5 Pós-processamento

As predições foram limitadas ao intervalo  $[0,5]$ , garantindo que os ratings previstos estejam dentro de um intervalo válido.

## 3 Resultados e Considerações

O sistema mostrou-se satisfatório em suas predições, mas há espaço para melhorias. O ajuste fino dos parâmetros, como taxa de aprendizado e regularização, pode ser feito de maneira mais sistemática. Uma limitação é o tempo de execução, que pode ser um obstáculo em aplicações em tempo real ou com conjuntos de dados muito grandes. Esse desafio ocorre pois o algoritmo implementado é  $O(m + n + e \times r \times k + t \times k)$ . As variáveis na análise de complexidade são definidas como:  $m$  é o tamanho da matriz  $P$  (número total de usuários únicos),  $n$  é o tamanho da matriz  $Q$  (número total de itens únicos),  $r$  é o número total de avaliações no conjunto de dados,  $k$  é o número de fatores latentes no funkSVD,  $e$  é o número de épocas e  $t$  é o número de linhas na matriz ‘targets’.

## 4 Conclusão

O sistema de recomendação desenvolvido oferece uma base sólida para a predição de ratings. Com ajustes e otimizações adicionais, é possível alcançar um desempenho ainda melhor e superar as limitações atuais.