



Research Challenge #1

Collaborative Product Recommendation

Arthur Pontes Nader - 2019022294

1) Introdução

Atualmente, os sistemas de recomendação possuem papel fundamental no comércio eletrônico e entretenimento digital, permitindo que empresas e plataformas ofereçam experiências personalizadas aos seus usuários. Tendo em vista a relevância da área, o objetivo deste trabalho foi implementar um sistema de recomendação de produtos baseado em algoritmos colaborativos.

Tentou-se diversas abordagens para solucionar o problema, tais como recomendação não personalizada (apenas para teste inicial), filtragem colaborativa usuário-usuário e item-item, fatoração de matrizes com viés de itens e usuários e até mesmo o algoritmo Mínimos Quadrados Alternados (ALS). Entretanto, o que realmente funcionou foi a tradicional fatoração de matrizes, que será explicada a seguir.

2) Implementação

Estruturas de Dados:

- Dataframes: O código faz uso da biblioteca Pandas para lidar com a entrada de dados. Ele lê avaliações e alvos a partir de arquivos CSV e realiza operações de limpeza e manipulação dos DataFrames.
- Fatores latentes: As matrizes P e Q são representadas como matrizes numpy, sendo que armazenam os fatores latentes dos usuários e itens, respectivamente.
- Dicionários: Um dicionário chamado "avaliacoes_usuarios" é usado para armazenar avaliações de usuários para itens, onde as chaves são tuplas (UserId, ItemId) e os valores são as classificações (Ratings). Além disso, há dicionários para mapear usuários e itens para, respectivamente, linhas e colunas das matrizes de fatores latentes.

Algoritmos:

- Fatoração de Matrizes: A fatoração de matrizes é a essência desse sistema de recomendação. O objetivo é aprender fatores latentes que representam

características de usuários e itens, sendo isso feito através das já mencionadas matrizes P e Q

- Stochastic Gradient Descent (SGD): O algoritmo de SGD é usado para otimizar os fatores latentes P e Q com base nas avaliações existentes. Ele itera sobre as avaliações, calcula os erros de previsão e atualiza os fatores latentes para minimizar esses erros. Os hiperparâmetros como taxa de aprendizado (alpha), regularização (reg), número de épocas (epocas) e quantidade de fatores latentes (K) devem ser devidamente ajustados para obter um bom desempenho. Essa função possui complexidade de tempo $O(\text{epocas} * |\text{avaliacoes_usuarios}| * K)$.
- Predição: Para fazer uma previsão de classificação para um usuário e item específicos, a função “predicao” calcula o produto interno dos fatores latentes correspondentes do usuário e do item em questão. Assim, essa função possui complexidade de tempo $O(K)$.

3) Resultados

No decorrer do desenvolvimento do trabalho, testaram-se 97 resultados de recomendações diferentes, tendo o último um score de 1.19886, obtido após 3 minutos e 35 segundos de execução. Os parâmetros utilizados durante o experimento foram os seguintes: $K = 225$, $\alpha = 0.002$, $\text{reg} = 0.05$ e $\text{epocas} = 20$. Esse resultado foi relativamente bom, sendo provável conseguir melhorá-lo por meio de testes com outras configurações para os hiperparâmetros.

Um dos insights mais importantes deste experimento foi a relação clara entre a quantidade de fatores latentes (K) e o desempenho do modelo. Os resultados indicam que, em geral, um aumento no número de fatores latentes levaram a uma melhoria significativa na qualidade das previsões.

Outro fator fundamental para a performance foi a inicialização das matrizes P e Q, de fatores latentes para usuários e itens. Esses valores foram iniciados de forma que o produto interno de uma linha de P por uma coluna de Q tivesse um valor esperado por volta de 4.02, que está dentro dos limites de avaliação que vão de 1 a 5 e perto da média das avaliações disponibilizadas. Esse modo de inicialização permite que o algoritmo consiga convergir para um ótimo mais rapidamente.

4) Conclusão

Os resultados obtidos permitem concluir que a implementação realizada foi satisfatória para solucionar o problema de recomendação de produtos para diferentes tipos de usuários baseado em uma abordagem colaborativa. Além disso, a implementação do algoritmo de fatoração de matrizes foi uma boa oportunidade de colocar em prática os conceitos vistos durante as aulas.