



Otimização de Hiperparâmetros para Modelos de Recomendação usando Algoritmos Genéticos

Matéria: Computação Evolucionista

Autor: Matheus Muniz Damasco

Professores: Heder Soares e Hélio José Correa





Introdução

O foco do meu projeto é a otimização de três hiperparâmetros em sistemas de recomendação, que são sistemas essenciais nos dias atuais. Diversas plataformas online como Netflix, iFood, Amazon, Medium e outras utilizam esses sistemas para aumentar suas vendas, engajamento e até o tempo de permanência dos usuários na plataforma. Dentro desse universo de sistemas de recomendação meu projeto aborda o Filtro Colaborativo (Collaborative Filtering), que utiliza as preferências de usuários com gostos similares para gerar recomendações personalizadas. No projeto, implementei dois modelos populares na área que são fatoração de matrizes e uma rede neural simples, ambos desenvolvidos com o Keras. O objetivo principal foi aplicar um algoritmo genético para otimizar três hiperparâmetros, buscando minimizar o erro absoluto médio (MAE) que foi a métrica de desempenho escolhida.



Descrição do Problema

- **Monoobjetivo:** Minimização do erro absoluto médio (MAE).
- **Algoritmos:** Fatoração de Matrizes e Rede Neural Simples.
- **Otimização (Hiperparâmetros):** Batch Size, Learning Rate, Latent Factors.
- **Variáveis do Projeto:** Batch Size (Quantitativa Discreta), Learning Rate (Quantitativa Contínua), Latent Factors (Quantitativa Discreta).
- **Restrições:** Limites definidos para o algoritmo de busca, valor mínimo e máximo de procura dos hiperparâmetros. Foram usados Batch Size entre 64 e 512, Learning Rate entre $1e-10$ e $1e-2$, Latent Factors entre 32 e 256. O erro absoluto médio (MAE) foi considerado após 10 épocas de treinamento em ambos os modelos.



Dados

Os Datasets escolhidos vem do MovieLens, administrados pelo GroupLens, um laboratório de pesquisa da Universidade de Minnesota. Pelo que eu li em artigos e pesquisas esses datasets são amplamente utilizados.

1. Movielens 100K (Data de Lançamento 04/1998):

- Contém 100.000 avaliações (Rating) variando de 1 a 5.
- Contém 943 usuários (Users).
- Contém 1682 filmes (Movies).

2. Movielens 1M (Data de Lançamento 02/2003):

- Contém 1.000.209 avaliações (Rating) variando de 1 a 5.
- Contém 6040 usuários (Users).
- Contém 3883 filmes (Movies).



Algoritmos

1. Low Rank Matrix Factorization (Keras):

Esse algoritmo é uma técnica amplamente utilizada em sistemas de recomendação. A ideia central é decompor a interação entre usuários e filmes em vetores de baixa dimensão (embeddings) que representam as características de cada um. Esse modelo é bastante eficiente para capturar preferências quando temos grandes volumes de dados e funciona muito bem com dados esparsos que é o nosso caso.

2. Neural Network (Keras):

Nossa Neural Network utiliza o princípio da Fatorização de Matrizes mas com o uso de Dropout, que ajuda a evitar overfitting ao remover conexões aleatoriamente durante o treinamento. Também usamos ativações como ReLU e Linear, o que torna o modelo capaz de aprender padrões mais complexos que o modelo de Fatorização de Matrizes sozinho não captaria.



Algoritmo Genético

Métodos de Seleção e Classificação:

- **Seleção por Torneio:** Um grupo de indivíduos são selecionados aleatoriamente da nossa população de acordo com o tamanho do torneio. Os indivíduos com o menor erro absoluto médio (MAE) entre os selecionados no torneio são escolhidos como pais para a próxima geração.
- **Fitness:** O fitness é definido como o menor erro absoluto médio (MAE) após 10 epochs do nosso algoritmo.
- **Elitismo:** Se o elitismo estiver ativado, 10% dos indivíduos (arredondados para cima, se necessário) com o menor erro absoluto médio (MAE) são preservados e diretamente transferidos para a nova geração, sem modificações.



População, Crossover e Mutação

- **População inicial:**

A população inicial do nosso algoritmo é gerada aleatoriamente, levando em conta os limites que aplicamos.

- **Crossover:**

Nosso método de crossover funciona da seguinte forma, definimos uma taxa e caso um nosso número aleatório esteja nesta taxa o crossover é aplicado, caso esteja fora não será aplicado. Os pais selecionados pelo método de crossover passam seu gene (hiperparâmetro) para seu filho de forma aleatória sendo 50% de herdar os genes de cada pai.

- **Mutação:**

Nossa mutação altera um gene (hiperparâmetro) aleatoriamente de acordo com o limite que definimos anteriormente.



Meus Parâmetros

- Número de Indivíduos: 50
- Número de Gerações: 10
- Tamanho do Torneio: 2
- Taxa de Crossover: 0.85 (85%)
- Taxa de Mutação: 0.15 (15%)
- Elitismo: True (10% dos indivíduos são passados para próxima geração)

Resultados:

Modelo	Dataset	Parâmetros Iniciais	MAE / Precision / Recall Inicial	Parâmetros Genéticos	MAE Genético/ Precision / Recall
Matrix Factorization	MovieLens 100K	Batch Size: 128, Learning Rate: 0.001, Latent Factors: 64	MAE: 2.1841, Precision: 0.875 Recall: 0.083	Batch Size: 226, Learning Rate: 0.0009, Latent Factors: 60	MAE: 1.0329, Precision: 0.843, Recall: 0.742
Matrix Factorization	MovieLens 1M	Batch Size: 128, Learning Rate: 0.001, Latent Factors: 64	MAE: 1.3042, Precision: 0.847, Recall: 0.572	Batch Size: 246, Learning Rate: 0.0005, Latent Factors: 57	MAE: 1.1226, Precision: 0.849, Recall: 0.671
Neural Network	MovieLens 100K	Batch Size: 128, Learning Rate: 0.001, Latent Factors: 50	MAE: 0.9327, Precision: 0.824, Recall: 0.999	Batch Size: 39, Learning Rate: 0.002, Latent Factors: 78	MAE: 0.9342, Precision: 0.835, Recall: 0.871
Neural Network	MovieLens 1M	Batch Size: 128, Learning Rate: 0.001, Latent Factors: 50	MAE: 0.9645, Precision: 0.843, Recall: 0.879	Batch Size: 34 , Learning Rate: 0.008, Latent Factors:78	MAE: 0.8726 Precision:0.836 Recall: 1.0



Conclusões

Os resultados mostram uma redução significativa no MAE após a aplicação da otimização, especialmente no modelo Matrix Factorization, com melhorias expressivas tanto no dataset MovieLens 100K quanto no 1M. Observamos também uma boa variação dos hiperparâmetros em relação aos valores iniciais mostrando que nossos algoritmos estão diversificando bem e explorando mais locais. Embora eu ter colocado precision e recall na tabela, o foco foi no MAE mas mesmo assim eles apresentaram melhorias após a otimização.



Referências

- [Grouplens \(Datasets\)](#)
- [A collaborative filtering recommendation system using genetic algorithm](#)
- [Fuzzy similarity optimization by genetic algorithm in user-based collaborative filtering recommender systems](#)
- [Recommenders Github](#)