Optimization for Collaborative Filtering

Abdelmalek RHAYOUTE

Certificat sciences des données et big data

February 23, 2023

Outline

Introduction

Collaborative Filtering is a popular approach in recommender systems, which recommends unseen items to users based on the preferences of similar users. In this project, we aim to use the gradient descent method to tackle the collaborative filtering problem.

Pre-process the dataset for cross-validation

We have 610 users, 9724 movies and the 100836 ratings matrix has 1.699968 percent of non-zero value.

The objective function

The matrix R is of size $m \times n$, and we are looking for $P \in R^{m,k}$ and $Q \in R^{n,k}$ such that $R \approx \hat{R} = PQ^T$. To do this, we consider the problem:

$$\min_{P,Q} \sum_{i,j:r_{ij}\text{exists}} \ell_{i,j}(R, P, Q),$$

where $\ell_{i,j}(R, P, Q) = \left(r_{ij} - q_j^\top p_i\right)^2 + \lambda \left(\|h(p_i)\|_2^2 + \|h(q_j)\|_2^2\right)$

and $(p_i)1 \leqslant i \leqslant m$ and $(q_j)1 \leqslant j \leqslant n$ are the rows of matrices P and

The parameter $\lambda \geqslant 0$ is a regularization parameter.

The function $h(a) = \min(a, 0)$

Implementing the gradient descent algorithm

The gradient of the function $h(x) = \min(x, 0)$ is not well-defined at $x = 0$, but we can still find a subgradient that satisfies the following inequality for all x:

$$\begin{cases} \{0\} & \text{if } x > 0 \\ [0, 1] & \text{if } x = 0 \\ \{1\} & \text{if } x < 0 \end{cases}$$

This subgradient can be used in gradient-based optimization methods replacing gradient descent

Using this subgradient, we can calculate the gradient of the objective function as follows:

$$\frac{\partial \ell_{i,j}}{\partial p_{i,k}} = -2q_{jk}(r_{i,j} - q_j^T p_i) + 2\lambda h'(p_{i,k})$$

$$\frac{\partial \ell_{i,j}}{\partial q_{j,k}} = -2p_{ik}(r_{i,j} - q_j^T p_i) + 2\lambda h'(q_{j,k})$$

where $h'(x)$ is the subgradient of $h(x)$.

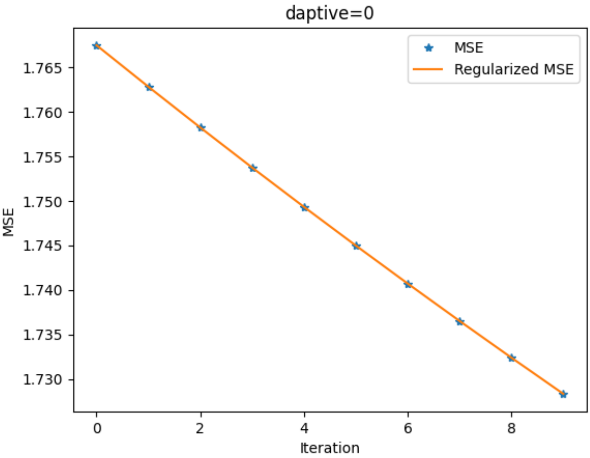
The gradient descent algorithm then updates the values of P and Q as follows:

$$p_{i,k} = p_{i,k} - \alpha \frac{\partial \ell_{i,j}}{\partial p_{i,k}}$$

$$q_{j,k} = q_{j,k} - \alpha \frac{\partial \ell_{i,j}}{\partial q_{j,k}}$$

where α is the learning rate. The algorithm continues to update the values of P and Q until the objective function converges to a maximum number of iterations is reached.

Compute RMSE score for the GD



Compute RMSE score for the the baseline method

Bibliography

https://github.com/Arhayout/outstanding-coursework-
https://lkpy.readthedocs.io/en/stable/bias.html