Optimization for Collaborative Filtering

Abdelmalek RHAYOUTE

February 23, 2023

Abstract

In this project, we aim to utilize the gradient descent method to tackle the collaborative filtering problem. We have access to a data file that holds the ratings of several movies from various users, which were collected by the MovieLens1 database. The challenge lies in the fact that not all users rate all movies, and thus, our task is to predict the missing ratings. The concept of collaborative filtering is to recommend unseen movies to a user by leveraging the ratings of other users. This approach enables us to make informed recommendations to a user based on their preferences, as well as the preferences of similar users.

1 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.

2 Solve the optimization problem to find an optimal P and Q

Having preprocessed the data, the next step is to develop a method to find the optimal P and Q on the training data. This will be achieved by:

- Calculating the objective and the gradient of the objective function, which will be the basis for our optimization problem.

  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 $(p_i)1 \leq i \leq m$ and $(q_j)1 \leq j \leq n$ are the rows of matrices P and Q respectively.

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

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

  The problem we are solving here is known as a "collaborative filtering" problem, which provides a possible solution to the Netflix problem.

- Implementing the gradient descent algorithm to find the optimal values of P and Q that minimize the objective function.

  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, such as 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 minimum or a maximum number of iterations is reached.

- Measuring the speed of this method to evaluate its performance and ensure it is efficient enough for large-scale problems.