CSCI 1952Q Coding 1

Introduction

This paper describes my approach to low-rank matrix completion. In particular, given k ratings (0.5 - 5.0) across m movies and n users in the MovieLens dataset, we want to predict q queries. My approach is based on matrix completion with non-convex optimization, and also includes an L2 regularizer and Adagrad optimizer.

Initialization

Firstly, we initialize a matrix $M \in \mathbb{R}^{n \times m}$ with the k given ratings. For a rating x from user $i \in [0, n-1]$ for movie $j \in [0, m-1]$, we set $M_{i,j} = x$. All unknown ratings are intitially set to 0.

Next, we perform Singular Value Decomposition (SVD) on M, expressed as $M = UDV^T$, where D is a diagonal matrix that includes the singular values of M. To achieve a rank-r approximation of M, we calculate $M = XY^T$, for $X \in \mathbb{R}^{n \times r}$ and $Y \in \mathbb{R}^{m \times r}$. Specifically, X is formed from $\tilde{U}\tilde{D}$, with $\tilde{U}$ being the matrix consisting of the first r columns of U, and $\tilde{D}$ being an $r \times r$ diagonal matrix with the top r singular values of M. Similarly, Y is derived from $\tilde{V}$, which contains the first r columns of V. We select $r = 24$ as our hyperparameter for this process.

Model Description

Our model operates by computing the matrix $A = XY^T$ at each iteration, where $A \in \mathbb{R}^{n \times m}$ represents the matrix of predictions. Each entry $A_{i,j}$ in this matrix corresponds to the predicted rating for the $(j+1)$th movie by the $(i+1)$th user. To ensure that all predictions fall within the realistic bounds of ratings between 0.5 and 5, any predictions outside this range are adjusted to the nearest boundary. However, after further evaluation, we opted to modify these limits to 0.75 and 4.75. This adjustment provides a more lenient margin for error, taking into account that a rating can only be excessively high or low, thereby spreading the potential error more evenly.

Objective Function

The objective function used in our model is defined as:

$$f(X,Y) = \left[\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}(M_{i,j}-A_{i,j})^2\right] + \lambda\left[\frac{1}{|S_X|}\sum_{(i,j)\in S_X} X_{i,j}^2 + \frac{1}{|S_Y|}\sum_{(i,j)\in S_Y} Y_{i,j}^2\right], \tag{1}$$

where M and A represent the matrices of actual and estimated ratings, respectively. Ω denotes the set of observed entries in the test set, and $|\Omega|$ is the count of these entries. The second term corresponds to an L2 regularization, averaging the squared entries of X and Y, scaled by $\lambda = 0.2$. Similar to Ω, S_X and S_Y represent the sets of all entries in X and Y, respectively, with $|\cdot|$ indicating the cardinality.

Training Procedure

The model is trained over 150 steps. At each step, we compute the gradient $\nabla f(X,Y)$ and update X and Y accordingly. Rather than using a standard stochastic gradient descent (SGD) method, our updates are based on an optimizer inspired by Adagrad, allowing for distinct learning rates for X and Y, providing a more nuanced adaptation of learning rates.

Validation

Validation loss is computed using the formula:

$$L = \frac{1}{|S|}\sum_{(i,j)\in S}(M_{i,j}-A_{i,j})^2, \tag{2}$$

where S is the set of entries held out for validation. With S comprising 10% of the test data, our model achieved a validation loss of approximately 0.6838, demonstrating its effectiveness in predicting unseen ratings within the validation set.