A Recommender System According to Collaborative Filtering

Jiageng Ding
Faculty of Engineering
University of Bristol
Bristol BS8 1UB, U.K.
lt22041@bristol.ac.uk

I. Recommender Systems Overview

Collaborative filtering(CF) is a common class of algorithms used in recommender systems(RS). Unlike content-based filtering algorithms, the core of collaborative filtering is to find similar users or similar items. Collaborative filtering can be divided into user-based CF (i.e. finding similar users), item-based CF (i.e. finding similar items) and matrix factorization CF.

A. Collaborative filtering by matrix factorization

In this experiment, we use matrix factorization(MF) collaborative filtering to construct a digital music recommendation system and perform sensitivity analysis. Collaborative filtering based on matrix factorization is to factorize the rating matrix R_{mn} into a user matrix P_{mk} and an item matrix Q_{nk}, which is further used to predict the missing ratings by the evolution of P and Q, i.e. $R_{predict} \approx P_{mk}Q_{nk}^T$.

$P_{mk} = [p_1, p_2, \ldots, p_m]$ in this expression denotes the projection of the user matrix on the K-dimensional space, and similarly $Q_{nk} = [q_1, q_2, \ldots, q_n]$ represents the projection of the items matrix on that space. This K-dimensional feature space is the latent feature of the rating matrix. Thus, the rating prediction of user i for item j can be approximated as the product of vector $\vec{p_i}$ and vector $\vec{q_j}$.

B. Matrix factorization based on SVD

The matrix P, Q can be generated by Singular Value Decomposition (SVD) of the scoring matrix. The sparsity of the actual scoring matrix (as in this experiment the percentage of missing values is 99.9%) makes this approach more difficult. Therefore, Simon Funk proposed a model to evolve the P, Q matrix by making the loss function gradually decrease. In this experiment, root mean squared error (RMSE) is used as the loss function, and a regularization term is added to avoid overfitting.

The optimization objective of this experiment is to find matrices P, Q that satisfy the following expressions,

$$min_{P,Q} \sum_{(p_i,q_j)\in TrainSet} (r_{ui} - p_i q_j^T)^2 + \lambda(\|p_i\|^2 + \|q_j\|^2)$$

II. Experimental Design

A. Data Set

This experiment uses digital music ratings as the data source, where the columns indicate user ID, music ID, ratings and time stamp respectively. The first 50,000 rows were selected from this as the experimental data set.

There are 34,947 unique users and 1,449 unique items in these 50,000 ratings.

B. Stochastic gradient descent

This experiment uses stochastic gradient descent (SGD) to update the parameters and thus reduce the loss function. Firstly, the gradient of the function is calculated as following(u,i is the training set with known ratings)

$$\frac{\partial L}{\partial p_u} = -2(r_{ui} - p_u^T q_i)q_i + 2\lambda p_u$$

$$\frac{\partial L}{\partial q_i} = -2(r_{ui} - p_u^T q_i)p_u + 2\lambda q_i$$

Update the vector p, q by SGD as

$$p_u = p_u + \alpha(2error_{ui} - 2\lambda p_u)$$

$$q_i = q_i + \alpha(2error_{ui} - 2\lambda q_i)$$

where α denotes the learning rate, and error denotes the error between the true rating and the predicted rating.

C. Training model

The default learning rate was set to 0.01 and the regularization parameter was set to 0.4. With the number of features set to 3 (k = 3), the training time is approximately 133 seconds when iterating through 40 epochs using the above SGD.

Figure 1 shows that the RMSE stabilises at around 0.9 at roughly 20 iterations.

III. Sensitivity Analysis

Firstly, a model training function is defined with 200 iterations for each value of k, and the RMSEs of the training set and validation set are output.

Next, the learning curves are plotted for k = (3, 8, 16, 64) respectively. As can be roughly seen in Figure 2, the RMSE for k=3 converges at 75 iterations, while the remaining three converge after 125 iterations.

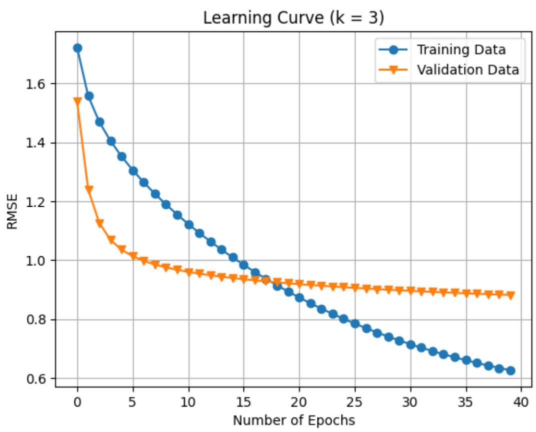

Fig. 1. Learning Curve for k=3 and epochs=40

Finally, the RMSEs of the last 100 epochs of each model were selected to plot the violin plot. Figure 3 clearly shows that the model with a feature number of 3 has the smallest mean RMSE and is the most stable.

Therefore, this paper considers k = 3 as the best parameter for this dataset and CF model.

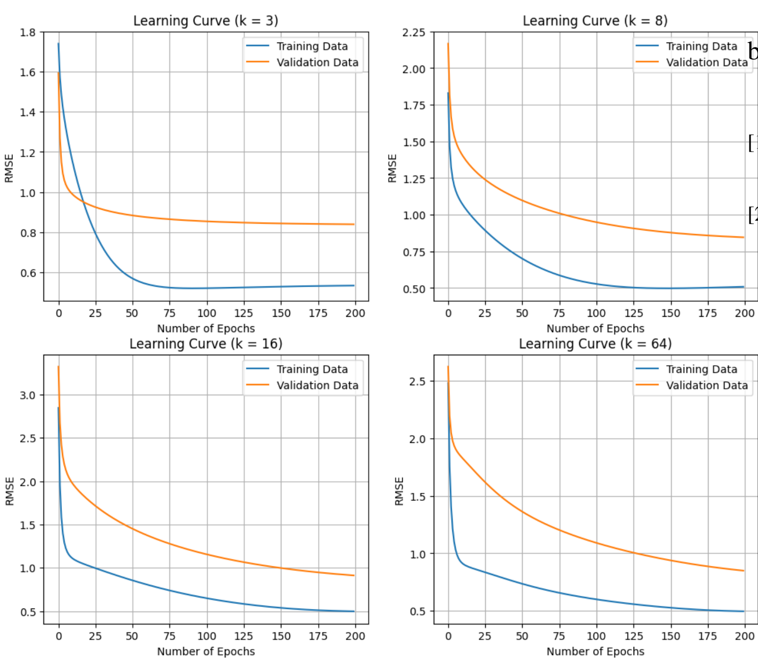

Fig. 2. Learning Curve for k=[3,8,16,64] and epochs=200

IV. Model Enhancement

A significant disadvantage for the CF algorithm is the inability to solve the sparsity of the initial data, i.e. the cold start problem. Many researchers have made enhancement strategies for this.

The solution proposed by Lika et al. in 2014, for example, is to group new users and find matching user data to make

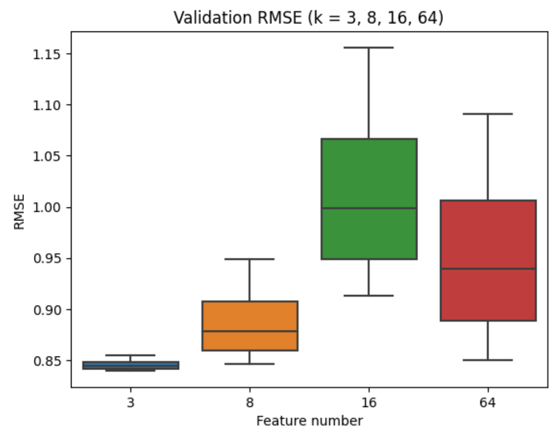

Fig. 3. Box plot for k=[3,8,16,64] and epochs=200

predictions. [1] Replacing the data of new users with data of other similar users is central to this.

A similar solution was proposed in the 2019 study by ikram et al. [2] Different groups of users were trained by using social information to segment different users. Through their study it can be seen that segmenting user groups can significantly improve the cold start problem.

There are also some solutions to the cold start problem based on greedy algorithms and neural networks.

References

[1] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, “Facing the cold start problem in recommender systems,” *Expert systems with applications*, vol. 41, no. 4, pp. 2065–2073, 2014.

[2] M. S. Ikram, A. W. Pillay, and E. Jembere, “Using social networks to enhance a deep learning approach to solve the cold-start problem in recommender systems.” in *FAIR*, 2019, pp. 173–184.