Assignment 3 - SVD Recommendation System

Team name: M2 Robo

Chan Kwan Yin	Lee Chun Yin	Chiu Yu Ying
3035466978	3035469140	3035477630
Team leader		

1. Background

Singular Value Decomposition (SVD) is a type of model-based collaborative filtering (CF) technique. It is based on matrix factorization (MF) to be an unsupervised learning method for latent variable decomposition and dimensionality reduction. It aims to learn the latent preferences of users and the latent factors of movies from ratings and then predict the target rating.

1.1. SVD with matrix factorization

It is in fact the another form of matrix factorization:

$$R = Q^T$$

We are able to truncate it into low rank k to approximately factorize as:

$$R \approx Q_k \sum_k P_k^T$$

We can further define from the above formula:

$$U = Q_k \sum_k$$

$$V = P_k$$

It is noticeable that the columns of U and V are mutually orthogonal and they can be used to represent users in a lower dimensionality.

1.2. Our SVD model

Inspired by Yehua Koren [1], we develop a SVD model in this assignment. The model predicts the rating r_{ui} indicating the preference for movie i by user u in the form:

$$\hat{r_{ui}} = \mu + b_u + c_i + q_i \cdot p_u$$

where the parameters to train are

- μ is the mean rating
- b_u is a user-specific bias
- c_i is a item-specific bias
- $\mathbf{q}_i$ and $\mathbf{p}_u$ are latent factors produced from matrix factorization with simple stochastic gradient descent (SGD) after the ratings are offset by the bias.They are representing the user and item characteristics respectively.

1.2.1 Detail of our model

To estimate the target ratings,we apply the following formula:

$$\hat{r_{u,i}} = \bar{r} + b_u + c_i + \sum_{f=1}^{F}(P_{u,f} \cdot Q_{i,f})$$

We decided to use the global average for $\bar{r}$ by

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} K_i$$

where K refers to the provided rating data.

Let us denote the prediction error e_{ui} = $r_{ui} - \hat{r_{ui}}$, Each biases and latent factors on known rating are the following:

$$b_u = b_u + \alpha \times (e_{ui} - \lambda \times b_u)$$

$$c_i = c_i + \alpha \times (e_{ui} - \lambda \times c_i)$$

$$P_{u,f} = P_{u,f} + \alpha \times (e_{ui} \times P_{u,f} - \lambda \times Q_{i,f})$$

where α is learning rate and λ is regularization term.

2. Technical Details

The training data set provided by Netflix consists of more than 100 million ratings with 17770 movies and 480189 users. Such a huge data set would consume a significant amount of training time and memory ($O(m^2n)$, since a correlation matrix between users is to be constructed), which is not possible for our hardware available. Therefore, only a subset of data is used for evaluation. To be specific, only first 10000 movies and first 1000 users that appear in the data set are considered.

2.1. Data preprocessing

The first 1000 movies are loaded into a numpy array with columns of Movie ID, User ID and Rating. The movie IDs and user IDs are reordered from 0 for the ease of indexing. Approximately 80% data are then reformatted into a rating matrix $R \in \mathbb{R}^{m \times n}$ for training, where r_{ij} is the rating of user i on movie j; the rest are retained for performance evaluation. The missing data are being skipped in our model.

2.2. Hyperparameters selection

In this SVD model, there are four hyperparameters to be selected, namely

- Learning Rate (α)
- Regularization (λ)
- Number of rank (k)
- Number of epoch

2.2.1 Learning Rate (α)

The learning rate indicates the speed at which the model learns. It controls how much to change the modal to respond the estimate error each time when the model weights are updated.

Rate with too small value may cause a long training time while a large value may allow the modal learning too fast to have less accurate or even not meaningful result.

2.2.2 Regularization (λ)

Regularization plays a crucial role in preventing the overfitting issue when training model. It aims to reduce the complexity of model to achieve its function. When the regularization parameter is large, the decay in weights during SGD update will be increased and therefore the weights of the hidden units will be negligible (close to zero).

If it sets to be too large, a large number of hidden units will be omitted and the network will loss its function. If it sets to be too small, the network will be too complex and hence will increase the possibility of being overfitted.

2.2.3 Number of rank (k)

Intuitively, it represents the classification of the user-movie data. In each category classified, the user U rating on movie A will also be grouped with the movie B when both movie A and movie B are considered as similar items (even when no record on the user U on movie B).

With a large value of it, we believe that the probability of having overfitting issue will be increased whereas the accuracy will drop with a low value on the number of rank.

2.2.4 Number of epoch

The number of epochs defines the number times that the modal will work through the whole training dataset.

Generally, we need to consider how the validation loss is behaving after each epoch. If the validation loss increases, it may be a sign of overfitting. The extremely high value of it will result in a long training time which harms its performance but may benefits the training process since it has more cycle of it. However,its effect on the model performance is not as significant as other parameters and we will not include it in the next section to present our model performance.

2.3. Predictive test set score (RMSE)

The model is evaluated by computing the RMSE between predicted and actual rating values:

$$\text{RMSE} = \sqrt{\sum_{(i,j)\in E} \frac{(\hat{r}_{ij} - r_{ij})^2}{|E|}}$$

where E is the set of retained evaluation data.

3. Model performance

The following table exhaustively lists our test results. Train RMSE and Test RMSE refer to the RMSE value calculated for the training dataset and testing dataset respectively.

rank=100 epoch = 0.2346 rank=50 epoch = 0.368

k	α	λ	Train RMSE	TestRMSE
10	1.234	1.234	1.234	1.234
10	1.234	1.234	1.234	1.234

References

[1] Y. Koren. Factorization meets the neighborhood: a multi-faceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008. 2