South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Xiongshuai Zhang
Jiachen Wang
Xuanyi Zhang

Supervisor:
Qingyao Wu

Student ID:
201530613641
201530612859
201530613658

Grade:
Undergraduate

December 27, 2017

Recommender System Based on Matrix Decomposition

Abstract—In this experiment we use Matrix Decomposition to construct a Recommender System ,which predict whether a user likes a movie he has not seen.

I. Introduction

RECOMMENDER System is a subclass of information filtering system that seeks to predict the *rating* or *preference* that a user would give to an item.

Recommender systems have become increasingly popular in recent years, and are utilized in a variety of areas including movies, music news, books, research articles. There are also recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life insurance, romantic partners, and Twitter pages.

Recommender systems typically produce a list of recommendations in one of two ways – through collaborative filtering or through content-based filtering (also known as the personality-based approach). Collaborative filtering approaches build a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.

In this experiment, we use Matrix Decomposition through collaborative filter to construct a simple Recommender System, and optimizes it by SGD.

II. Methods and theory

1. Identify set of items rated by the target user
2. Identify which other users rated 1+ items in this set (neighborhood formation)
3. Compute how similar each neighbor is to the target user (similarity function)
4. In case, select k most similar neighbors
5. Predict ratings for the target user's unrated items (prediction function)
6. Recommend to the target user the top N products based on the predicted ratings

Target user u, ratings matrix Y
yv; i rating by user v for item i Similarity Pearson r correlation sim(u; v) between users u&v

$$sim(u,v) = \frac{\sum_{i\in I_{u,v}}(\mathbf{y}_{u,i}-\hat{\mathbf{y}}_u)(\mathbf{y}_{v,i}-\hat{\mathbf{y}}_v)}{\sqrt{\sum_{i\in I_{u,v}}(\mathbf{y}_{u,i}-\hat{\mathbf{y}}_u)^2\sum_{i\in I_{u,v}}(\mathbf{y}_{v,i}-\hat{\mathbf{y}}_v)^2}}$$

Predicted rating $\mathbf{y}^*_{\mathbf{u,i}}$

$$\mathbf{y}^*_{\mathbf{u,i}} = \hat{\mathbf{y}}_u + \frac{\sum_{j\in I} sim(\mathbf{v}_j,\mathbf{u})(\mathbf{y}_{v_j,i}-\hat{\mathbf{y}}_{v_j})}{\sum_{j\in I}|sim(\mathbf{v}_j,\mathbf{u})|}$$

Algorithm's process :

- *Given* learning rate α and λ
- *Initalize* $P^{(1)},\ldots,P^{(n_u)},Q^{(1)},Q^{(n_m)}$ to small values.
- *Minimize* $J(P,Q)$ using SGD : for every $i = 1,..,n_u, j = 1,\ldots,n_m$:

$$P_{k(j)} -= \alpha\Big(\sum_{i:r(i,j)=1}\big((P_{(j)})^T Q_{(i)} - y_{(i,j)}\big)Q_{(ki)} + \lambda P_{k(j)}\Big)$$

$$Q_{k(i)} -= \alpha\Big(\sum_{j:r(i,j)=1}\big((P_{(j)})^T Q_{(i)} - y_{(i,j)}\big)P_{k(j)} + \lambda Q_{(ki)}\Big)$$

- For a user with parameters θ and a movie with features x, predict a rating of $\theta^T x$.

III. Experiments

A. Dataset

- Utilizing MoviesLens-100k dataset.
- u.data – Consisting 10,000 comments from 943 users out of 1682 movies, At least, each user comment 20 videos, Users and movies are numbered consecutively from number 1 respectively. The data is sorted randomly.
- u1.base / u1.test are train set and validation set respectively, separated from dataset u.data with proportion of 80% and 20%. It also make sense to train set and validation set from u1.base / u1.test to u5.base / u5.test.

B. Implementation

- 1. Read the data set and divide it. Populate the original scoring matrix R_{n_u,n_m} against the raw data, and fill 0 for null values.
- 2. Initialize the user factor matrix $P_{n_u,n}$ and the item (movie) factor matrix $Q_{n_m,n}$, where n is the number of potential features.
- 3. Determine the loss function and hyperparameter learning rate α and the penalty factor λ.
- 4. Use the stochastic gradient descent method to decompose the sparse user score matrix, get the user factor matrix and item (movie) factor matrix:

- 4.1 Select a sample from scoring matrix randomly;
- 4.2 Calculate this sample's loss gradient of specific row(column) of user factor matrix and item factor matrix;
- 4.3 Use SGD to update the specific row(column) of $P_{n_u,n}$ and $Q_{n_m,n}$;
- 4.4 Calculate the L_{val} on the validation set, comparing with the L_{val} of the previous iteration to determine if it has converged.

- 5. Repeat step 4. 40000 times, get a satisfactory user factor matrix P and an item factor matrix Q, Draw a L_{val} curve with varying iterations.
- 6. The final score prediction matrix is $R_{nu,nm}$ obtained by multiplying the user factor matrix $P_{nu,n}$ and the transpose of the item factor matrix $Q_{n_m,n}$.

TABLE I: Simulation Parameters

learning rate α	0.01
λ	0.00001
number of features n	10

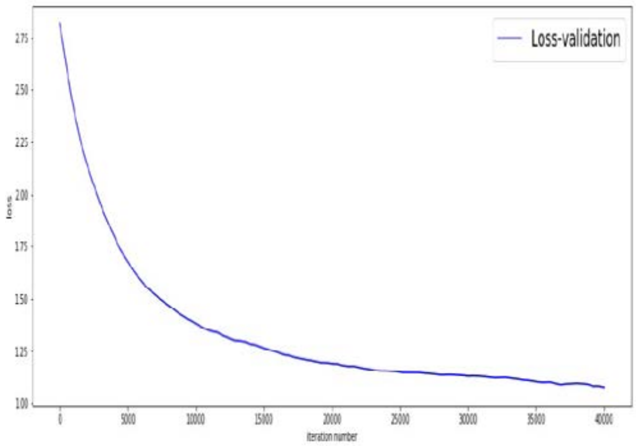

Fig. 1: L_{val} curve with varying iterations.

IV. Conclusion

We successfully constructed a basic recommender system through collaborative filter. We choose SGD to converge the loss and get the best result as we can. Recommend systems have been widely used so it's necessary for us to understand how it works, and realize it ourselves. Collaborative filter need the participance of other users, so the more users use this system, the more accurate the result will be.