# Processing Massive Data Project – Report

Liu Jundi – 5120309623
Gao Hongyuan – 5120309624
Gao Lei – 5120309583
Gui Qichao - 5120309629

## 1. Introduction

Large-scale recommendation system is a crucial part of many websites such as Netfilx. The main idea of recommendation is to predict the unobserved data using the information that we have already observe. And this job is sometimes known as the matrix completion problem. In this project, we implement a movie recommendation system and give predictions to the given test set. Clearly, it is a typical matrix completion problem of movie rating predictions. So in the implementation phase, we used the traditional regularized singular value decomposition method (RSVD). Nevertheless, there are some particular special features of this project, such as the users' tastes and the movies' genres would change merely along the record time and so on. Therefore, we modified the original model to fit our dataset properly in order to ensure the predictions precisely.

## 2. Regularized Singular Value Decomposition

Firstly, we briefly introduced the singular value decomposition (SVD). With the knowledge of linear algebra, we know that every matrix has a SVD:

$$A = U\Sigma V^T$$

Where A is an m*n matrix, U is an m*m orthogonal matrix, $\Sigma$ is a m*n diagonal matrix and V is a n*n orthogonal matrix. The basic idea of matrix completion problem is based on SVD. More precisely, we will construct a user-movie matrix which each row represents a distinct user, each column represents a different movie and each entry corresponding to the unique rating.

In order to fit the model with the particular problem, we will make a slight modification in the prediction function:

$$\hat{r}_{ui} = \mu + b_i + b_u + p_u^T q_i$$

Where $\mu$ is the global average rating, $b_i$ is the movie property, $b_i$ is the

user property.

But this model is not practical because if we use the least square method to optimize the cost function it may cause overfitting. So finally, we introduce the regularization term to avoid overfitting. The cost function is as below:

$$\frac{1}{2}\sum_{u,i}[r_{ui}-(\mu+b_i+b_u+p_u^T q_i)]^2+\frac{1}{2}\lambda\sum_u|p_u|^2+\frac{1}{2}\lambda\sum_i|q_i|^2+\frac{1}{2}\lambda\sum_u|b_u|^2+\frac{1}{2}\lambda\sum_i|b_i|^2$$

where $\lambda$ is the regularized coefficient.

To solve this least square equation, we use stochastic gradient descent method which the update equation is as below:

$$p_{uk} := p_{uk}+\eta(e_{ui}q_{ki}-\lambda p_{uk})$$

$$q_{ki} := q_{ki}+\eta(e_{ui}p_{uk}-\lambda q_{ik})$$

$$b_u := b_u+\eta(e_{ui}-\lambda b_u)$$

$$b_i := b_i+\eta(e_{ui}-\lambda b_i)$$

## 3. Data Processing

The training set has about 20 million lines of rating record, which is composed of user's ID, movie's ID, date ID and rating and we read them into 4 arrays respectively. For the sake of constructing matrix P and Q, the program iterates for tens of times till the decrease of error is less than a pre-defined threshold.

## 4. Experiment

- Environment

  Operation system: Ubuntu 14.04, Mac OS X 10.10

  Language: Python 2.7

- Evaluation

  During the practical experiment, we adjust learning rate and lambda by observing the result. However, constrained by the speed of the language of Python and our CPU, we didn't set a very high dimension and threshold. The final test file we submitted which achieved an RMSE as much as 0.93 has the learning rate of 0.01, multiplied by 0.95 for every iteration, and lambda is set as 0.005 and the dimension is 13.

  We also tried to run an SVD++ method to improve our performance, but the large amount calculation of y (which leads to an 8-hour-long iteration) stopped us from doing that despite using libraries like NumPy.

## 5. Conclusion

We implement an SVD-based Python program to predict a user's potential rating of a movie. We achieved the RMSE of 0.93, which is not very good due to computing ability. We had to admit that under such circumstance a lower-level programming language such as C may be a better choice. In the future, we'll try to migrate the project to other languages.

Many thanks to Mr. Zhou Jingyu, Mr. Ye Run and Mr. Zhou Peng.