The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Xingyu Chen, Qichen Huang,
Huakui Zhang

Supervisor:
Mingkui Tan

Student ID:
201530611289, 201530611777,
201530613610

Grade:
Undergraduate

December 28, 2017

Recommender System Based on Matrix Decomposition

***Abstract*—In this experiment, we try to explore the construction of recommender system and understand the principle of matrix decomposition, which will make us more familiar with the use of gradient descent. While Constructing a recommendation system under small-scale data set, we will be able to cultivate our engineering ability.**

I. Introduction

A 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, search queries, and products in general. Recommender systems typically produce a list of recommendations in one of two ways through collaborative filtering or through content-based filtering. Collaborative filtering approaches build a model from a user's past behaviour 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. Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties. In this experiment, we use the model-based collaborative filtering based on matrix decomposition with the method of stochastic gradient descent in order to implement a movie recommender system.

II. Methods and Theory

A. Matrix Decomposition

Matrix decomposition or matrix factorization is a factorization of a matrix into a product of matrices.

Give a rating matrix $R \in \mathbf{R}^{m,n}$ with sparse ratings from m users to n items. Matrix decomposition factorizes rating matrix R into the multiplication of two low-rank feature matrices $P \in \mathbf{R}^{m,k}$ and $Q \in \mathbf{R}^{k,n}$. See Fig. 1.

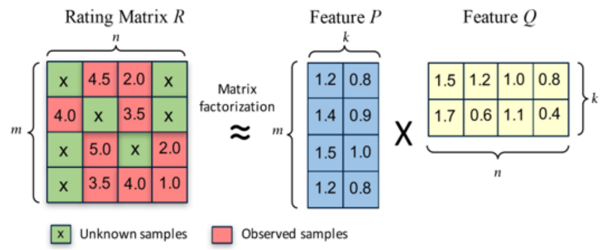

Fig. 1. Matrix Decomposition

B. Stochastic Gradient Descent)

Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. In other words, SGD tries to find minima or maxima by iteration. In this experiment, we use the cost function

$$J(x^1, ..., x^{n_m}, \theta^1, ..., \theta^{n_u}) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2.$$

The gradient of the loss function is

$$\frac{\partial J}{\partial x_k^i} = \sum_{j:r(i,j)=1} ((\theta^i)^T x^i - y^{i,j}) \theta_k^j.$$

$$\frac{\partial J}{\partial \theta_k^i} = \sum_{i:r(i,j)=1} ((\theta^i)^T x^i - y^{i,j}) x_k^i.$$

where x is the feature vector of every item, θ is the feature vector of every user and r is a accessorial matrix(if a user i scores the movie$j$, then $r(i,j)$ in the matrix r is 1, otherwise 0).

III. Experiments

A. Dataset

This experiment uses MovieLens-100k dataset in which u.data consists 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, seperated from dataset u.data with proportion of 80% and 20%. It also makes sense to train set and validation set from u1.base / u1.test to u5.base / u5.test.

B. Implementation

In this part we will describe how we do our experiment in details.

1. Read the data set and divide it (or use u1.base / u1.test to u5.base / u5.test directly). Populate the original scoring matrix R_{n_users,n_items} against the raw data, and fill 0 for null values.

2. Initialize the user factor matrix $P_{n_users,K}$ and the item (movie) factor matrix $Q_{n_items,K}$, where K is the number of potential features.

3. Determine the loss function and hyperparameter learning rate η 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 $P_{n_users,K}$ and $Q_{n_item,K}$;

4.4 Calculate the $L_{validation}$ on the validation set, comparing with the $L_{validation}$ of the previous iteration to determine if it has converged.

5. Repeat step 4. several times, get a satisfactory user factor matrix P and an item factor matrix Q, Draw a $L_{validation}$ curve with varying iterations.

6. The final score prediction matrix $\hat{R}_{n_users,n_items}$ is obtained by multiplying the user factor matrix $P_{n_users,K}$ and the transpose of the item factor matrix $Q_{n_items,K}$.

We initialize the user factor matrix $P_{n_users,K}$ and the item (movie) factor matrix $Q_{n_items,K}$ with random floating numbers in [0,1] and set number of potential features $K = 100$, learning rate $\alpha = 0.1$ and regularized parameter $\lambda = 10$.

We implement both batch gradient descent(BGD) and stochastic gradient descent(SGD), and run it for 300 and 30000 iteration respectively, with results shown in the Fig. 2 and Fig. 3.

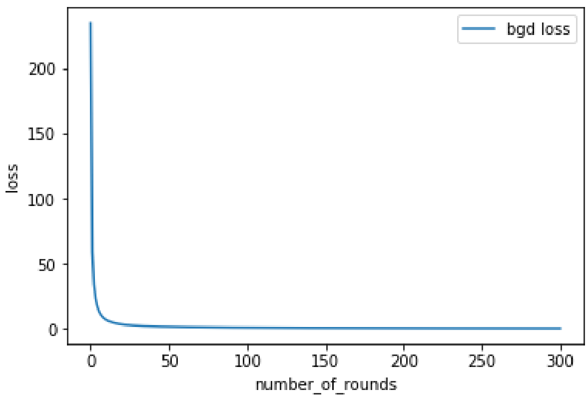

Fig. 2. BGD Loss Figure.

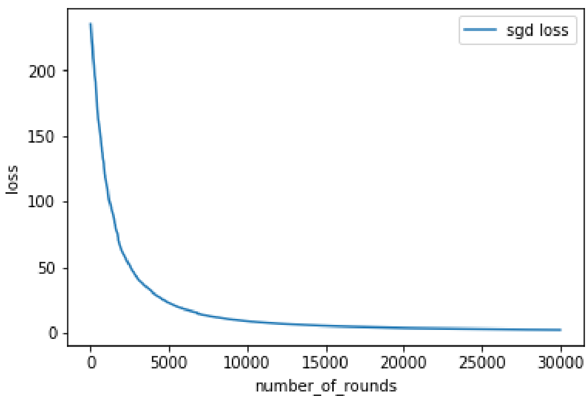

Fig. 3. SGD Loss Figure.

From Fig. 2 and Fig. 3, we can see that the loss converges after rounds using the BGD method and converges after 15000 rounds using the SGD method, both indicating the success of the implementation of the movie recommender system. And it is also convenient to draw a conclusion that BGD converges faster than SGD though consumes more unnecessary resources sometimes.

IV. Conclusion

In this experiment, we have explored and implemented a movie recommender system and better understood the principles of matrix decomposition. This experiment swept the haze of our minds and to some extent reveals the business logic of some enterprises like Taobao, which greatly triggers our ambition to dig more deeply into the recommender systems.