Movie Recommender System

Faisal Usman(P14-6016), Mussawer Al Yasah(P14-6105)

Abstract

In this project, a movie recommender system is built based on the MovieLens 10M dataset. We used collaborative filtering method to predict user's movie rating and we can recommend movies to customers, which they potentially give high ratings according to prediction. The root-mean-square error (RMSE) is calculated to carry out evaluation.

Introduction

Recommender systems are used to provide personalized recommendations according to user profile and previous behavior. Recommender systems are widely used in the Internet Industry. Services like Amazon, Netflix, and YouTube are typical examples of recommender system users. Recommender systems cannot only help the users find their favorite products, but also bring potential profit to online service providers.

Dataset

The MovieLens 10M is used as dataset in our project. The MovieLens 10M dataset contains 10,000,054 ratings for 10681 movies from 71,567 users. Each user has more than 20 ratings. The ratings for each movie are from 1 to 5. This dataset is randomly divided into 2 parts: the training set and the test set. For each user, the training set contains 90% of the user's ratings. The rest 10% ratings build up the test set. Collaborative filtering is trained based on the training set and algorithm evaluation is carried out based on the test set.

Problem Formulation

Recommendation is currently a very popular application of machine learning. In our project, we are trying to recommend movies to customers. We use the following definitions:

n_u = number of users

n_m = number of movies

$r(i, j) = 1$ if user j has rated movie i

$y(i,j)$ = rating given by user j to movie i (defined only if $r(i,j) = 1$)

$\theta^{(j)}$ = parameter vector for user j

$x^{(i)}$ = feature vector for movie i

For user j, movie i, predicted rating: $(\theta^{(j)})^T (x^{(i)})$

Method

In our project, we used collaborative filtering method to predict user's movie rating and we can recommend movies to customers, which they potentially give high ratings according to prediction.

Collaborative Filtering

To get the parameter for all users, we do the following:

$$\min_{\theta^{(1)},\dots,\theta^{(n_u)}} = \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} ((\theta^{(j)})^T (x^{(i)}) - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

However, it can be very difficult to find features in a movie. To figure this out, we use feature finders:

$$\min_{x^{(1)},\dots,x^{(n_m)}} = \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} ((\theta^{(j)})^T (x^{(i)}) - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} (x_k^{(i)})^2$$

To speed things up, we can simultaneously minimize our features and parameters:

$$J(x,\theta) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

It looks very complicated, but we've only combined the cost function for theta and the cost function for x. Because the algorithm can learn them itself, the bias units where $x_0 = 1$ have been removed, therefore $x \in \Re^n$ and $\theta \in \Re^n$

Algorithm Implementation

There are three steps in the algorithm:

1. Initialize $x^{(i)},\dots,x^{(n_m)},\theta^{(1)},\dots,\theta^{(n_u)}$ to small random values.
2. Minimize $J(x^{(1)},\dots,x^{(n_m)},\theta^{(1)},\dots,\theta^{(n_u)})$ using gradient descent.
   E.g. for every $j = 1,\dots,n_u$, $i = 1,\dots n_m$:
   $x^{(i)} := x^{(i)} - \alpha\{ \quad ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})\theta^{(j)} +$
   $\lambda x^{(i)} \}$ $k\ k\ k\ k$
   $\theta^{(j)} := \theta^{(j)} - \alpha\{ \quad ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})x^{(i)} + \lambda\theta^{(j)} \}$ k
   $k\ k\ k$
3. For a user with parameters θ and a movie with (learned) feature x, predict a star rating of $\theta^T x$.

Mean Normalization

If the ranking system for movies is used from the previous lectures, then new users (who have watched no movies), will be assigned new movies incorrectly. Specifically,

they will be assigned θ with all components equal to zero due to the minimization of the regularization term. That is, we assume that the new user will rank all movies 0, which does not seem intuitively correct.
We rectify this problem by normalizing the data relative to the mean. First, we use a matrix Y to store the data from previous ratings, where the ith row of Y is the ratings for the ith movie and the jth column corresponds to the ratings for the jth user.

We define a vector $\propto = [\propto_1, \propto_2, \ldots, \propto_{n_m}]$, such that $\propto_i = \frac{{}_{j:r(i,j)=1} Y_{i,j}}{{}_j r(i,j)}$ then normalize the data by subtracting $\propto$ from the actual ratings for each user.
The new linear regression prediction including the mean normalization term is $(\theta^{(j)})^T x^{(i)} + u_i$

Algorithm Evaluation

After the algorithm is trained, root-mean-square error (RMSE) is used to evaluate the algorithm. The RMSE is defined as:

$$\sqrt{\frac{{}_{(i,j)\in T}(r_{ij} - r_{ij})}{N}}$$

r_{ij} , r_{ij}^* are the actual rating and predicted rating of user j on movie j respectively.

Where T is the test set, N = |T| is the number of movie ratings of test set. i, j are the index of movie and user respectively.
From figure 1, we can conclude that prediction performance becomes better with the increasing of number of iterations.

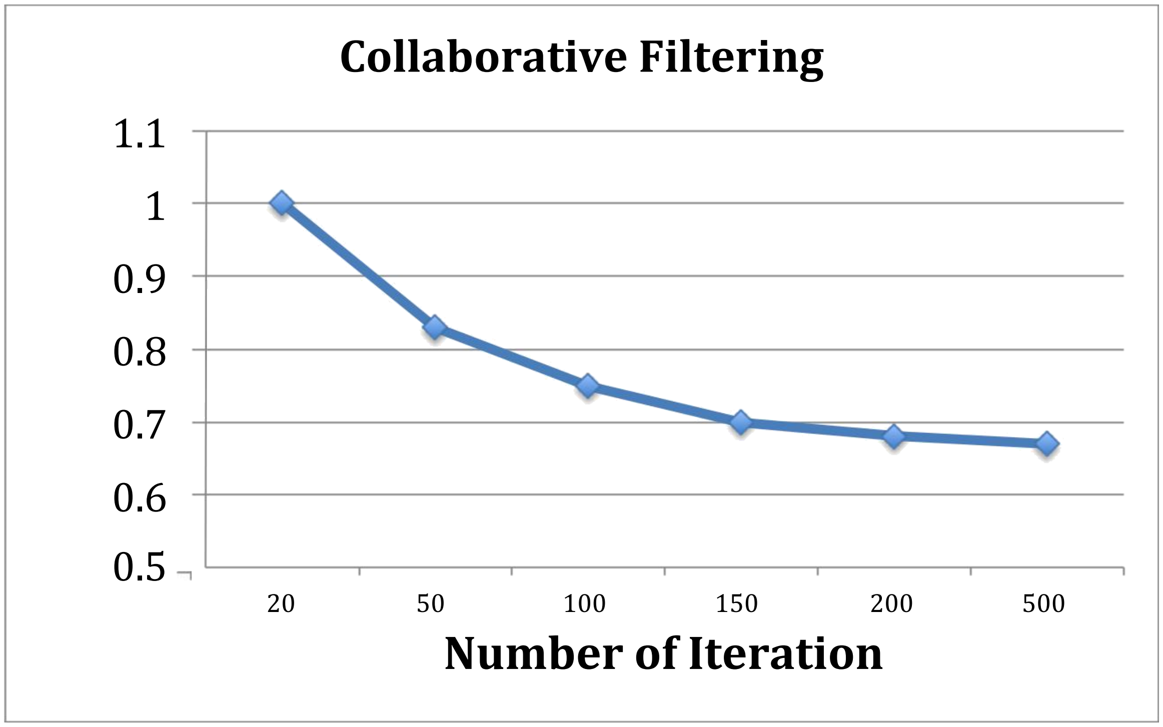

Figure 1

Conclusion

In our project, collaborative filtering algorithm is used to predict user's movie rating. The MovieLens dataset, which has 10 million ratings, is selected in our project and divided into training set and test set. The RMSE method is used for algorithm evaluation. According to evaluation result, our movie recommender system has pretty good prediction performance.

Reference

[1] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42.8 (2009): 30-37.

[2] Bao, Zhouxiao, and Haiying Xia. "Movie Rating Estimation and Recommendation." CS229 (2012).

[3] Ricci, Francesco, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. Springer US, 2011.

[4] Rajaraman, Anand, and Jeffrey David Ullman.Mining of massive datasets. Cambridge University Press, 2011.