Movie Recommender System

Yijie Zhuang, Boyang Xu, Hao Wu, Shaoyi Han, Hong Jin

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.

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 is 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: $(\theta^{(j)})^T(x^{(i)})$

Collaborative Filtering

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

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

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

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

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

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

Algorithm Implementation

These are the steps in the algorithm:

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

Mean Normalization

In order to recommend movies to new customers, who have watched no movies, we rectify our model by normalizing the data relative to the mean.

Define a vector $\mu=[\mu_1,\mu_2,\dots,\mu_{n_m}]$, such that $\mu_i=\frac{\sum_{jr(i,j)=1}Y_{i,j}}{\sum_j r(i,j)}$

Then normalize the data by subtracting μ *from the actual ratings for each user.*

The new linear regression prediction including the mean normalization term is $(\theta^{(j)})^T x^{(i)}+u_i$

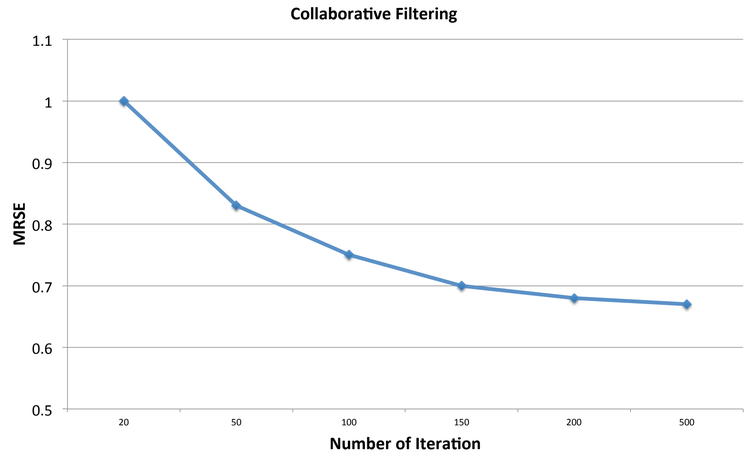
Algorithm Evaluation

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

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

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.



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.