A framework for collaborative filters

Yanping Xie, Zuoyue Li, Jie Huang
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—Collaborative filters help people evaluating items based on the opinions of other people. In this paper, we propose several different collaborative filter methods and build a common hybrid framework to combine those techniques for a final recommendation. In traditional collaborative filters, it is usual to apply only single algorithm. In our systems, it is flexible for extending and mixing new models, which can quickly produce high quality recommendations.**
We conduct four methods: regularized singular value decomposition(RSVD), K-means, improved RSVD and Postprocessing SVD with kernel ridge regression. And we use linear model to weight four models. The framework gains a XXX% improvement comparing to the SVD baseline in the movie dataset.

I. Introduction

Recommender systems are widely used in nowadays life, which help us choose books, movies, music and even friends. In collaborative filter, the recommendation is based on the preference of the user towards other items and the opinion of others[— add some reference later—]. In many traditional collaborative filters work, they use only single model which is limited in some specific application. For example, some methods perform quite well in the book recommendation but is weak in the music recommendation[— add some reference later—]. In this paper, we propose a hybrid framework to mix various recommendation algorithms.
The rest of the paper is organized as follows. In the section 2, we first introduce regularized singular value decomposition(RSVD), K-means, improved RSVD and Postprocessing SVD with kernel ridge regression which are used in our system. Then we describe how we combine those four methods. In the section 3, we present our experiment design and some results. In the section 4, we give some conclusions.

II. Models and Methods

A. Problem Formalization

In this paper we focus on the movie rating problem. We are given some training samples including user i, movie j and their corresponding ratings r_{ij}. All ratings are integer values between 1 and 5 stars. And we should evaluate the ratings for new pairs user i' and movie j'.

B. Basic Model

1) Regularized singular value decomposition(RSVD): We map both users and movies to a joint latent factor space of dimensionality k. Accordingly each user i is associated with a vector $u_i \in \mathbf{R}^k$, while each movie j is associated with a vector $v_j \in \mathbf{R}^k$. The dot product $u_i^T v_j$ captures the interaction between user i and movie j. We use this product to estimate the rating of user i and item j.

$$\hat{r_i j} = u_i^T v_j$$

To learn the factor of the user and item, the system minimizes the regularized squared error on the set of known ratings.

$$\min_{u,v} \sum_{i,j \in kn} (r_{ij} - \hat{r}_{ij})^2 + \lambda(||u_i||^2 + ||v_j||^2)$$

where $(||u_i||^2 + ||v_j||^2)$ is the regular terms to avoid overfitting and kn is the set of pair (i,j) of which rating r_{ij} is known.

2) Improved regularized singular value decomposition: In the RSVD methods, we use dot product to fit the real rating. However, it is shown that some biases exist among both users and items. For example, some users tend to give higher rate than others. So we introduce the bias term into the evaluation.

$$\hat{r_i j} = u_i^T v_j + b_i + b_j + \mu$$

where b_i and b_j are the biases of user and items. μ is the average of all rating. The system learns by minimizing the squared error function:

$$\min_{u,v} \sum_{i,j \in kn} (r_{ij} - \hat{r}_{ij})^2 + \lambda(||u_i||^2 + ||v_j||^2 + b_i^2 + b_j^2)$$

3) K-means: K-means algorithm is used to divide users into K clusters C_k and minimizing the intra-cluster variance.

$$\sum_{k=1}^{K} \sum_{i \in C_k} (||r_i - \mu_k||^2)$$

, where

$$||r_i - \mu_k||^2 = \sum_{j \in kn_i} (r_{ij} - \mu_{kj})^2$$

, where kn_i is the set of movie rated by user i. For each user i in cluster k, we use μ_{kj} to rate movie j. Besides we compute the average 11 runs of K-means with $K \in$ [4, 24] and stride 2 as the prediction.

4) Postprocessing SVD with kernel ridge regression: .

[bu shi hen dong a xiong di !] .

Predictor	RMSE/Linear/SGD	RMSE/Nolinear/FGD	RMSE/linear/FGD
SVD	1.007	0.0	0.0
RSVD	1.020	0.0	0.0
RSVD2	1.009	0.0	0.0
K-means	1.046	0.0	0.0
RSVD-with krr	1.068	0.0	0.0
RSVD2-with krr	1.080	0.0	0.0
Linear Model	1.018	0.0	0.0
Merge Model	0.978	0.0	0.0

Table I
RMSE OF DIFFERENT MODEL

C. Improvement

1) Full Gradient vs Stochastic Gradient: In order to minimize the error of mean square, Stochastic Gradient Descent(SGD) is widely used [—— reference——]. But SGD is easily influenced by noise data, although it owns fast convergence speed. So we implement both full gradient decent and stochastic gradient decent and compare them in the experiment part.

2) Linear Interaction vs Nolinear Interaction: Either RSVD and improved RSVD adopt linear Interaction between user and items. We conduct a nolinear function f in the interaction part.

$$\hat{r_i j} = f(u_i^T v_j)$$

$$\hat{r_i j} = f(u_i^T v_j) + b_i + b_j + \mu$$

where we use $f(x) = 4 * sigmod(x) + 1 in our experiment$

D. Ensemble

III. Experiment

A. Datasets

We use a movie rating data set to evaluate variants of item-based recommendation algorithms. The datasets include ratings of 10000 users for 1000 different movies. All ratings are integer values between 1 and 5 stars.The number of known rating is 1176952 and the average is 3.857. And we split the datasets into two part: 90% training set and 10% test set.

B. Metric

Our collaborative filtering algorithm is evaluated according to the following weighted criteria: prediction error, measured by root-mean-squared error (RMSE)

C. Experimental Results

As shown in the table I,

1) Full Gradient vs Stochastic Gradient:

2) Linear Interaction vs Nolinear Interaction:

3) single model vs ensemble model:

IV. Summary

A. Conclusion

B. Future Work

ACKNOWLEDGEMENTS

The author thanks Christian Sigg for his careful reading and helpful suggestions.