A Hybrid Framework for Collaborative Filtering

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

***Abstract*—Collaborative filtering (CF) evaluates users' preference over unrated items taking advantage of the ratings of other users and items. In this project, we implemented multiple collaborative filtering methods including memory-based ones and model-based ones, and combined them into one hybrid framework which is flexible for extension. The methods we adopted includes: the basic statistics, the user-based neighbourhood-based method, K-means, Singular Value Decomposition (SVD) with dimension reduction, the regularized SVD, the regularized SVD with bias, the kernel ridge regression based on SVD and a linear weighted model. The final hybrid framework was obtained via fitting the Ridge regression on all the predictors and the two-way interactions between some of them. The framework gained a 2.56% improvement compared with the SVD with dimension reduction baseline on the Kaggle public dataset.**

I. Introduction

Nowadays, recommender systems are widely used in our daily life. They can help us choose books, movies, music or even friends. In collaborative filtering (CF), the recommendation of an item to a user can done using the user's preference towards other items and the opinion of other users towards this item [— add some reference later—]. Many traditional collaborative filtering works have proposed good algorithms and models [—reference—], but using only a single method has its limitation. (For example???) So in this project, we implemented a hybrid framework that mixes various CF models and achieves better performance than any single one of them. Both memory-based models and model-based models were exploited and the results were merged via Ridge regression.

The rest of the paper is organized as follows. In Section 2, we first introduce the CF models used in our system, including the basic statistics, the user-based neighbourhood-based method, K-means, Singular Value Decomposition (SVD) with dimension reduction, the regularized SVD, the regularized SVD with bias, the kernel ridge regression based on SVD and a linear weighted model. Then the details of how we combine the methods via Ridge regression are introduced. In Section 3, the experiment design and results will be presented. Conclusions are given in Section 4.

II. Models and Methods

A. Problem Formalization

In this paper, we are focusing on the movie rating problem. We are given some training samples which are composed of the user id i, movie id j and the corresponding rating r_{ij}. All the ratings are integer values between 1 and 5. The task is to predict the rating $r_{i'j'}$ given user i' and movie j'.

B. Models

We have implemented both memory-based methods and model-based methods, including the user-based neighbourhood-based method, the SVD with dimension reduction and some of the methods introduced in [—ref_a—].

1) Basic Models: Same to [—ref_a—], we used 6 basic predictors which exploit the simple statistics of the data.

2) User-based Model: In the User-based collaborative filtering, the prediction of one user's rating to an item is an linear combination of the ratings of this item provided by the users similar to him. We implemented a method described in [—ref_b—]. The similarity between users is evaluated via the Pearson correlation coefficient.

$$Pearson formula$$

For a prediction of user i and item j, no more than K most similar users who have rated item j will be selected and their ratings of item j will be averaged with the weight being their Pearson coefficient to user i.

3) 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.

$$\hat{r_{ij}} = u_i^T v_j$$

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.

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

4) 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.

$$\hat{r_{ij}} = u_i^T v_j + b_i + b_j + \mu$$

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:

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

5) 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.

6) Postprocessing SVD with kernel ridge regression: In the Postprocessing SVD with kernel ridge regression(PSVD-KRR), we discard all user factor vector u_i after training. For each user we predict the rating via regression of v_j. For user i, vector y denotes the all real rating $r_i$in training data. And X denote a matrix of movie features. We can use kernel ridge regression to predict y.

$$\hat{y_i} = K(x_i^T, X)(K(X, X) + \lambda I)^{-1} y$$

where $K(X, X')$ is a kernel function. We use $K(x_i^T, x_j^T) = exp(2(x_i^T x_j - 1))$ in our experiments.

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.

$$\hat{r_{ij}} = f(u_i^T v_j)$$

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

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

D. Ensemble

We employ a linear model to ensemble all models. The w_m denotes the weight of mth model we want to combine and $\hat{r_{ij}^m}$ is the predicted results of model m for user i and item j. The algorithm wants to training the weight w by minimize

$$\sum_{i,j \in kn} (r_{ij} - \sum_{m \in M} w_m \hat{r}_{ij}^m)^2 + \lambda(||w_i||^2)$$

where M is the set of model we using.

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

We implement six models: SVD, RSVD, Improved RSVD(shown as RSVD2), RSVD with KRR, RSVD2 with KRR and K-means in the experiment. We compare different training methods(FGD/SGD) and different interaction function(linear/nolinear) on those six models. Besides, we compare using single model and the ensemble models. All results are summarizes in the table I.

1) Full Gradient vs Stochastic Gradient: As shown in the second and third column, FGD performs better than SGD. On the model XX, FGD obtain an XXX% improvement on XXmodel(XXX%) than SGD.

2) Linear Interaction vs Nolinear Interaction: The Nolinear Interaction acquire a better results than linear interaction on all models. More specifically, it has a largest improvement on XXmodel(XXX%), while has limited effect on XX model (XXX%).

3) Single model vs Ensemble model: The ensemble model that combines all six above-mentioned model achieve a XXX RMSE. The ensemble gain a XXX improvement compare to XXX, which is the best model among all single model. The results demonstrate that the ensemble methods strengthen the ability of model to fit complicated data.

All experiments were done on the ETH cluster euler with XXXGHz processor and XXXGB RAM. Running times varied from XXXXmin for SVD KNN to around XXX for RSVD2.

IV. Summary

A. Conclusion

We introduce various basic collaborative filters algorithms and build a framework to combine them all. Besides, we compare different training methods and interaction functions to gain a better movie rating. The experiments show out that an ensemble model with FGD and nolinear interaction model achieve a best results(XXX %)

B. Future Work

1) Model: We may try to combine some deep learning models in our framework.

2) Data: Now we just limit our model on movie rating datasets. More experiments are needed on other recommendation tasks.

Acknowledgements

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