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 nonlinear 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, the nonlinear biased regularized SVD and some of the methods introduced in [1].

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

2) User-based Model: In the User-based (UB) 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 by the Pearson correlation coefficient:

$$Pearson(u,v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u) \cdot (r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}}$$

, where μ_u and μ_v are the mean ratings of user u and user v, and I_u and I_v denote the set of indices of movies rated by user u and v respectively.

For a prediction of user u and item j, no more than K most similar users who have rated item j will be selected and the set of these similar users is denoted by $P_u(j)$. Their ratings of item j will be averaged with the weight being their Pearson coefficient to user u. Before the averaging, the raw ratings will be mean-centered in a user-wise fashion in order to tackle the problem that different users might provide ratings on different scales. The mean rating of user u will be added back to the weighted average to get the final prediction.

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} Sim(u,v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_u(j)} |Sim(u,v)|}$$

K is set to 500 in our system.

3) K-means: We used the K-means algorithm to classify the users into K clusters C_k and to minimized the intra-cluster variance. The intra-cluster variance is defined as in

[1],

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

, 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 a user i classified to cluster k, μ_{kj} is our predicted rating for the movie j. We have run the K-means algorithm 11 times with K varying from 4 to 24 with a stride of 2 and took an average of the predicting results as our prediction.

4) SVD with Dimension Reduction: In the SVD with dimension reduction method, we keep the k most significant singular values for the reproduction of the rating matrix.

$$A = U[:, k] \cdot D[: k, : k] \cdot V[:, k]^T$$

For each movie j, the missing values in its column are filled with the mean of its observed ratings. k is set to 12 in our system.

5) Regularized SVD: In the regularized SVD (RSVD) model, both the users and movies are mapped to a joint latent factor space with dimension k. Each user i is associated with a vector $u_i \in \mathbf{R}^k$, and each movie j is associated with a vector $v_j \in \mathbf{R}^k$. The dot product $u_i^T v_j$ which captures the interaction between user i and movie j is used to estimate the rating of item j by user i.

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

To learn the user and item vectors, we implemented both the stochastic gradient descent method and the full gradient descent method aiming at minimizing the squared error. A regularization term is added in order to avoid over-fitting.

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

where $(||u_i||^2 + ||v_j||^2)$ is the regularization term and kn is the set of observed pairs (i, j). For the stochastic gradient descent version, $k = 10$, $initiallearningrate = 0.01$, $\lambda = 0.02$ and for the full gradient version, $k = 32$, $initiallearningrate = 0.1$, $\lambda = 0.1$.

6) Biased Regularized SVD: Biases exist among both users and items. For example, some users tend to give higher rating than others. So in the biased regularized SVD (BRSVD) method, two bias terms are introduced.

$$\hat{r_{ij}} = u_i^T v_j + c_i + d_j + \mu$$

where c_i and d_j are the bias terms for user i and item j respectively. μ is the mean of all observed ratings. The objective function is the same as in the RSVD method. Both the stochastic gradient descent version and the full gradient descent version were implemented. For the stochastic gradient descent version, $k = 5$, initial learning rate = 0.01, $\lambda = 0.02$ and for the full gradient version, $k = 32$, initial learning rate = 0.1, $\lambda = 0.1$.

7) Nonlinear Regularized SVD with Bias: The nonlinear regularized SVD with bias (NSVD) further considered adding some nonlinearity onto the biased regularized SVD, as introduced in [—red_c the blog—]. The ratings are modeled by the function

$$r_{ij} = 4 * (sigmoid(u_i^T v_j) + b_i + b_j) + 1$$

The training objective is the same as that of RSVD. The parameters are learned using full gradient descent. $k = 32$, learning-rate-like parameter = 400, $lambda = 0.02$.

8) Kernel Ridge Regression Based on SVD: In the kernel ridge regression (KRR) based on SVD model [1], the user vectors u_i are discarded and the movie vectors v_j are used as predictors. For a user i, the target vector y contains the observed ratings r_i in the training set and X denotes a matrix of movie features where each row of X is the normalized movie vector v_j. Kernel ridge regression is trained 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. $K(x_i^T, x_j^T) = exp(2(x_i^T x_j - 1))$ is used in our system. $k = 32$, $alpha = 0.7$.

9) Linear Model: The linear weighted model (LW) introduced in [1] is also adopted in our system. In this model, each movie j has a weight w_j. The rating of a movie j by a user i is linear to the sum of the weights of the user's rated movies.

$$\hat{r}_{ij} = m_j + e_i \cdot \sum_{j_2 \in J_i} w_{j_2}$$

, where J_i is the set of movies rated by user i, m_j is the mean rating of movie j and constant weights $e_i = (|J_i| + 1)^{-1/2}$ The movie weights are learned via gradient descent which minimizes the regularized square error. The learning rate is set to 0.01 and the regularization parameter is set to 0.02.

C. Ensemble

In order to take advantage of different CF models and overcome their individual shortcomings, we decide to combine the results of the models. Ridge regression was fitted on the validation set to obtain the ensemble weights. w_m denotes the weight of the mth mode and $\hat{r_{ij}^m}$ is the predicted results by model m for user i and item j. Ridge regression trained the weights w_m by minimizing the regularized square error

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

where M is the set of models we implemented and. The two-way interactions of some of these models were also fed to the ridge regression as features.

To avoid the possible drawback caused by the small size of the validation set (which is the training set for the ridge regression), we generated two training/validation data splits from the original training data. Ridge regression was run on each split and then the regression results were merged using averaging.

III. Experiment

A. Data

The data set is a movie rating data set that is composed of 10000 users and 1000 movies. The rating values are integers between 1 and 5. There are in total 1176952 observed ratings and the mean of the observed ratings is 3.857.

The observed ratings was split into a training set with approximately 90% of the observations and a validation set with approximately 10% of the observations. Two training/validation splits were generated from the original data. The training set was used for the training of individual CF models and the validation set was used for the validation of the individual CF models and the training of the ensemble regression.

B. Metric

The performance of the methods and models is measured by the root-mean-squared error (RMSE).

$$RMSE = \sqrt{\frac{1}{|P|} \cdot \sum_{(i,j) \in P} (\hat{r}_{ij} - r_{ij})^2}$$

, where P is the set of user/movie pairs to be tested.

C. Experiment Settings

All experiments were run on the ETH cluster Euler with XXXGHz processor and XXXGB RAM. The runnings time varied from 2min for SVD with dimension reduction to several hours for the user-based method and models using full gradient descent.

D. Experimental Results

We have implemented six basic predictors and nine CF models: UB, K-means, SVD, RSVD, BSVD, NSVD, KRR based on RSVD, KRR based on BSVD and L-W. Experiments on individual models were conducted and the ensemble result is also reported. Different training methods(FGD/SGD) and different interaction functions(linear/nolinear) were compared. The ensemble strategy has also been assessed.

1) Individual Results: The RMSEs on the validation set of different CF methods are shown in Table I. The content in the brackets indicates that the training method is stochastic gradient descent (SGD) or full gradient descent (FGD). KRR-RSVD denotes the KRR method based on RSVD and KRR-BSVD denotes the KRR method based on BSVD.

The regularized SVD with bias trained with FGD achieved the best score. The user-based method and the non-linear regularized SVD with bias also had good performance.

Model	RMSE
UB	0.99968
K-means	1.04551
SVD	1.00649
RSVD (SGD)	1.01988
RSVD (FGD)	1.00266
BSVD (SGD)	1.00901
BSVD (FGD)	0.98544
NSVD	0.99913
KRR-RSVD	1.06843
KRR-BSVD	1.08037
LM	1.01836

Table I
RMSE of Different Models on Validation Set

2) Full Gradient Descent vs Stochastic Gradient Descent: For the RSVD model and the BSVD model, we have trained them using the stochastic gradient descent method and the full gradient descent method. As we can see from Table I, FGD consistently performs better than SGD (1.69% better for RSVD and 2.34% better for BSVD).

3) Linear Interaction vs Nonlinear Interaction: The NSVD method adds nonlinearity to the interaction between user vectors and movie vectors comparing with BSVD. We expected the nonlinearity will increase the model complexity and improve the prediction quality. However, as shown in Tabel I, the RMSE of NSVD is no better than that of BSVD. One possible reason is that the training of NSVD in our experiments has not converged due to its relatively slow training speed. Given more training time or better hardware, it's hopeful that NSVD will outperform BSVD.

Strategy	RMSE
Ridge + 2-way + 2 splits	0.97937
Ridge + 2 splits	0.98038
Linear + 2-way + 2 splits	0.97996
Ridge + 2-way	0.98168

Table II
RMSE of Different Ensemble Strategies on Kaggle Public Set

4) Ensemble: The final system uses the six basic predictors and eight of the CF models: UB, K-means, SVD, RSVD (SGD), BSVD (SGD), KRR-RSVD, KRR-BSVD and LW. All the predictors and models have been trained on two training/validation data splits. Ridge regression was run on each splits to obtain the corresponding ensemble weights. The final system takes the average of the two regression results and the RMSE on the Kaggle public set is shown in Table II with the row name "Ridge + 2-way + 2 splits". The RMSE score of the BSVD (FGD), which is the best single model on the validation set, is 0.98611 on the Kaggle public set. The hybrid system gained a 0.68% improvement over the best single model we have implemented. This result demonstrates that the ensemble method can benefit from the advantages of different models and achieve a better

prediction ability.

We also tested different ensemble strategies, including one that didn't add two-way interactions of the models, one that used linear regression instead of ridge regression and one that used only one training/validation data split. The RMSE scores of these strategies are all higher than our final strategy (ridge regression with two-way interaction on two data splits).

IV. Summary

A. Conclusion

In this project, we implemented various collaborative filtering models and built a hybrid framework that combines them all. The ensemble of different CF methods benefits from the advantages of these models and can overcome their individual limitations. We have conducted comprehensive experiments to tune the models to their best performance and tested different ensemble strategies. The system that combines the basic predictors, the user-based method, K-means, SVD with dimension reduction, the regularized SVD, the regularized SVD with bias, the kernel ridge regression based on RSVD and BSVD, the linear weighted model and the two-way interactions between some of them using ridge regression on two training/validation data splits achieves the best RMSE score, which is 0.97937 on the Kaggle public set.

B. Future Work

We may incorporate some deep learning models into our framework. And for now we have only evaluated our model on the movie rating data set. More experiments can be conducted on other recommendation tasks to test the generalization ability of our system.

References

[1] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," *Proceedings of Kdd Cup Workshop*, 2007.