Available online: November 02, 2018

Commun.Fac.Sci.Univ.Ank.Series A2-A3
Volume 60, Number 2, Pages 15-26 (2018)
DOI: 10.1501/commua1-2_0000000112
ISSN 1303-6009 E-ISSN 2618-6462
http://communications.science.ankara.edu.tr/index.php?series=A2A3

AN ENSEMBLE MODEL FOR COLLABORATIVE FILTERING TO INVOLVE ALL ASPECTS OF DATASET

YILMAZ AR

ABSTRACT. The accuracy of predictions is better if the combinations of the different approaches are used. Currently in collaborative filtering research, the linear blending of various methods is used. More accurate classifiers can be obtained by combining less accurate ones. This approach is called ensembles of classifiers. Different collaborative filtering methods uncover the different aspects of the dataset. Some of them are good at finding out local relationships; the others work for the global characterization of the data. Ensembles of different collaborative filtering algorithms can be created to provide more accurate recommender systems.

1. INTRODUCTION

Recommender systems make predictions about the users' preferences and based on these predictions recommend specified items or services to their members. Collaborative Filtering is one of the most popular technologies in recommender systems [1]. The two most important user tasks within the collaborative filtering context are "annotation in context" and "finding good items" [1]. Collaborative filtering is widely used by commercial web sites like Amazon, Netflix, and Last.fm to recommend items to their visitors or members. In collaborative filtering past-user behavior is very important and the prediction is almost totally depends on these behaviors like purchasing some products, rating items or services. The most popular collaborative filtering approaches are neighborhood based ones. The taste or preferences of a user is determined by using the recorded preferences of the likeminded users. In these methods the key point is finding the likeminded users in other words neighbors. Latent Semantic Analysis is other kind of methods used in the collaborative filtering research. Let's assume that there is a dataset of user

Received by the editors: October 17, 2018; Accepted: October 30, 2018.
Key word and phrases: Collaborative filtering, ensemble methods, negative correlation learning.

ratings for a specified number of distinct items. This dataset can be considered as a user-item matrix. The purpose is to predict the rating of a given user on given items by using known ratings. The known ratings can be factored into latent features by using linear algebra [2]. The dataset are broken into principal components [2]. Using this information, the prediction of a user's rating on an item can be made [2]. Singular Value Decomposition is one of the best factorization methods. Besides these methods, there are other kinds of approaches for collaborative filtering; some of them will be discussed in Related Work section.

2. Related Work

2.1 Work Related To Collaborative Filtering

Amazon.com developed its own algorithm called item to item collaborative filtering to personalize its web site to each customer's interest [3]. In this method the approach tries to find similar products instead of similar users. The main advantage is scalability. In Amazon.com example the number of customers is much more than the number of products therefore the methods which uses similar customers do not fit well.

In [4], Bell and Koren discussed some global effects. Let's give some examples of these effects; there are some systematic tendencies for some users to specify higher ratings than others. This situation is also applicable for items; some products get higher ratings than others. Another one is; some users' or some products' ratings may suddenly or slowly change over time [4]. Removing these effects improves the accuracy of neighborhood based collaborative filtering [4].

The neighborhood based techniques in local scale and SVD-like matrix factorization in regional scale was used in [5]. In same study also the combination of local scale and regional scale approaches in a unifying model was presented.

In [6], the method of neighborhood-aware matrix factorization was discussed. The neighborhood information was included in regularized matrix factorization model [6]. The procedure which is called "slot blending" is used as an ensemble method.

Three different matrix factorization (MF) approaches; regularized MF, maximum margin MF and non-negative MF was used in [7]. In this work, a simple ensemble scheme was utilized for collaborative filtering. All three methods were applied and later the average of the multiple predictions was calculated as the final prediction value [7].

In [9], the method of "imputation boosted collaborative filtering" was presented. Populating the sparse user-item matrix by using an imputation technique is the first step of this method. Later a traditional Pearson correlation-based collaborative filtering algorithm is executed on imputed data. Koren suggested a new neighborhood based model that aims to optimize a global cost function [10]. He also integrated implicit feedback to SVD-based latent factor model [10].

In study [25], a genetic algorithm solution to collaborative filtering is provided. User-to-user similarity weights are refined using genetic algorithms before they are used in prediction process.

The work [12] pointed out three drawbacks of memory-based collaborative filtering methods. These are; the methods are sometimes over optimistic in making a prediction, they do not consider side information and they usually provide unjustifiable inferences [12]. This work proposes there features that a memory-based prediction algorithm should take into account: appropriate similarity measure metric, individual prediction and user's preferences and rating patterns [12].

2.2 Work Related To Ensemble Methods

In 1990, a technical report is published which was about the task decomposition through modular connection architecture. Maybe this study was the foundation of current ensemble techniques. In this study, a novel modular connectionist architecture was proposed. In this approach, in order to learn the training data, the networks that are placed in the architecture need to compete [16]. After this competition, dissimilar training patterns were learnt by different individual networks. Therefore, different networks were be able to calculate different functions [16]. The architecture gets the capability of partitioning a main job into two or more independent functions. After that different individual networks are assigned to learn each function [16].

Free energy minimization with variation is used to infer the parameters of a mixture of experts' model using ensemble learning [15]. This method avoids the problem of over fitting [15].

Methods for constructing ensembles were studied in [17]. They are; Bayesian voting: hypothesis enumerating, training example manipulation, input feature and output target modification, and inserting randomness [17]. Ensemble of classifiers can often perform better than any individual classifier [19].

To create classifiers in an ensemble, bagging was used in [11]. It produces them by getting random samples from the original dataset with replacement. For each sample, only one classifier was produced [11].

Bagging actually uses the concept of random and independent manipulations on training data applied by bootstrap sampling [11]. On the other hand the boosting strategy recommends guided manipulations of the training data [11].

An ensemble classifier produced by using attribute selection and diversity measure is studied in [20]. In this study, the method works as follows: By using a random attribute set, a classifier is learnt. Later, the diversity measure between the created classifier and all the current ensemble members is computed [20]. If the diversity is sufficient enough, the classifier joins to the ensemble; otherwise it is discarded [20].

Categorization scheme identification and the description of the multiple classifier systems were provided in [13]. In this study, the types of multi-net systems are categorized with respect to some issues. That systems may use bottom up or top down approach. If they use bottom up method, they might apply either static or fixed combination approach. They may unite either ensemble, modular or hybrid parts. These systems may consist of competitive or cooperative combination methods [13].

In [14], the error diversity is discussed; different heuristic and qualitative explanations in the literature are provided. To create diverse ensembles, the various techniques would be used. They are surveyed in [14].

2.3. Work Related To Negative Correlation Learning

Ensemble learning via negative correlation was discussed in [23]. All the individual networks that are placed in the ensemble are trained simultaneously and interactively using the correlation penalty terms in their error functions in negative correlation learning [23]. NCL can produce networks to improve specialization and cooperation among the individual networks [23].

In the study of [24], the evolutionary ensembles with NCL were presented. It was demonstrated that the effectiveness of the algorithm is related to the size and the complexity of the ensemble [21]. The theoretical links between the well-known regression ensemble and a linearly combined classifier ensemble was studied in [22].

3. Problem Description

There are a specified number of users and movies. The ratings of the users on specified movies on a specified date are given. There are also additional information about movies and users. The release dates of movies and movie videos, the genre of movies, user age, user occupation and user Zip code are some information that can be used.

The aim is to predict the rating of a specified user on a specified movie on a given date.

4. Algorithm

First of all, we decided which of the information was used in this study. We created a data set consisting user age, user gender, 19 movie-genres field, timestamps and the ratings. We eliminated the user ids and item ids because of the difficulties on normalization issues. We do not believe the movie release dates, video release dates, movie titles and IMDB URL information would much help therefore they all were eliminated from our study. Because of the computational power and time constraints we did not use the occupation and Zip code information.
After this deciding process, our dataset consists of user age, user gender, 19 movie-genres field, timestamp and rating. The last field is target one. We normalized the fields: age, timestamp, and rating by using the following formula:

$$x_{normalized} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{4.1}$$

These fields were normalized between 0.0 and 1.0. The other fields gender and 19-fields genre are already normalized taking the value of 0 or 1.

MovieLens dataset has five different partitioning; 80 000 for training and 20 000 for testing purposes. We used one of them in our study.

After this preprocessing we had a training dataset with 22 input-field and 1 target-field with the amount of 80 000. We had a testing dataset with the amount of 20 000.

We created a neural network that has one hidden layer with 4 nodes. In this network we had 22 input nodes and one output node. We used the logistic sigmoid function as the activation function.

$$\text{output}=\frac{1.0}{1.0+\exp(-\text{net})} \tag{4.2}$$

We used standard back propagation algorithm. A momentum constant is used in weight updating process. Our program executed with different values of learning rate and momentum constant.

We created a simple ensemble of three neural networks. The output of the ensemble is just averaging the individual neural networks outputs.

$$F(n)=\frac{1}{\mathrm{M}}\sum_{i=1}^{M}F_i(n) \tag{4.3}$$

M shows the number of the individual neural networks in given ensemble. $F_i(n)$ is the output of neural network *i* on the *n*th training pattern. The output of the ensemble on the *n*th training pattern is represented by $F(n)$ [24].

We also created another ensemble consisting again three neural networks but trained using negative correlation learning. The output of the new ensemble is also the averaging the individual networks outputs but the individual networks are trained simultaneously.

The idea of neural network ensembles with negative correlation learning is to encourage different individual NNs in the ensemble to learn different parts or aspects of the training dataset. Negative correlation learning introduces a correlation penalty term into the error function of each individual network in the ensemble. With the help of this term, the networks in the ensemble learn the training data simultaneously and interactively.

$$\begin{aligned}E_{\mathrm{i}}&=\frac{1}{N}\sum_{n=1}^{N}E_i(n)\\&=\frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(F_i(n)-d(n))^2+\frac{1}{N}\sum_{n=1}^{N}\lambda p_i(n)\end{aligned} \tag{4.4}$$

is a correlation penalty function. The main goal here is to correlate each networks' error negatively with the other networks in the ensemble [24]. λ takes the values between 0 and 1. It is used to set the degree of the penalty.

$$p_i(n) = (F_i(n) - F(n)) \sum_{j \neq i} (F_i(n) - F(n)) \tag{4.5}$$

The partial derivative on the *n*th training example, with respect to the output of network *i* is:

$$\frac{\partial E_i(n)}{\partial F_i(n)} = (1 - \lambda)(F_i(n) - d(n)) + \lambda(F(n) - d(n)) \tag{4.6}$$

5. Dataset

MovieLens dataset was used in this study. The dataset comprised of 100 000 ratings from 943 users on 1682 movies. Ratings were given between 1 and 5. It was downloaded from http://www.grouplens.org/taxonomy/term/14. In this dataset at least 20 movies were rated by each user. Some demographic information for the users was included.

Three main files in the set were u.data, u.item, and u.user.

u.data consists of 100 000 ratings by 943 users on 1682 items. Users and items are numbered consecutively from 1. This data is randomly ordered. The time stamps are Unix seconds since 01.01.1970 UTC. The set is a tab separated list of: user id, item id, rating, timestamp.

u.item includes information about the items (movies). The first 5 fields in the list are of the item id, item title, item's release date, video release date, and IMDB URL. The remaining 19 fields represent the genre of the item that is a movie in that dataset. 0 shows that the movie is not of that genre, on the other hand 1 indicates it is of the given genre. The movies con be in more than one genre at once. Movie ids in this set are the ones as item ids in u.data. The following genres are used in that dataset in given order: unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western.

u.user has the demographic information about the users. The set is a tab separated list of: user id, age, gender, occupation, zip code.

6. Results And Conclusion

The RMSE results of the first experiment are given in Table 1.Given results are also shown in Figure 1. In that experiment, the simple ensemble in which three networks are trained individually and compared with NC ensemble that trains three networks simultaneously. The parameter negative correlation constant (*ncc*) in that experiment is set as 0.9.

The RMSE results of the second experiment are given in Table 2.Given results are also shown in Figure 2. In that experiment, the simple ensemble in which three networks are trained individually and compared with NC ensemble that trains three networks simultaneously. The only difference with the first experiment is ncc parameter that is set in second one as 0.99.

RMSE (Root Mean Square Error) (101 Epochs)		
Randomization Seed	Simple Ensemble (Individual training of three networks) lr: 0.05 mc:0.9	NC Ensemble (Simultaneous training of three networks) lr: 0.05 mc:0.9 ncc:0.9
1	1.107711434	1.087123451
5	1.111880434	1.131227910
10	1.112127800	1.081843158
15	1.111352544	1.085239553
20	1.110780784	1.085402535
50	1.109321873	1.112112262
100	1.116207508	1.144173996
200	1.118746130	1.146657720
300	1.117772735	1.086918043
400	1.112187941	1.151998674
500	1.113877294	1.136682508
600	1.122937664	1.091284358
700	1.101775745	1.129200684
800	1.110991583	1.087093331
900	1.110193667	1.135609214
1000	1.117178851	1.090603402

Table 1. RMSE of Two Different Ensembles (ncc is 0.90 in second ensemble)

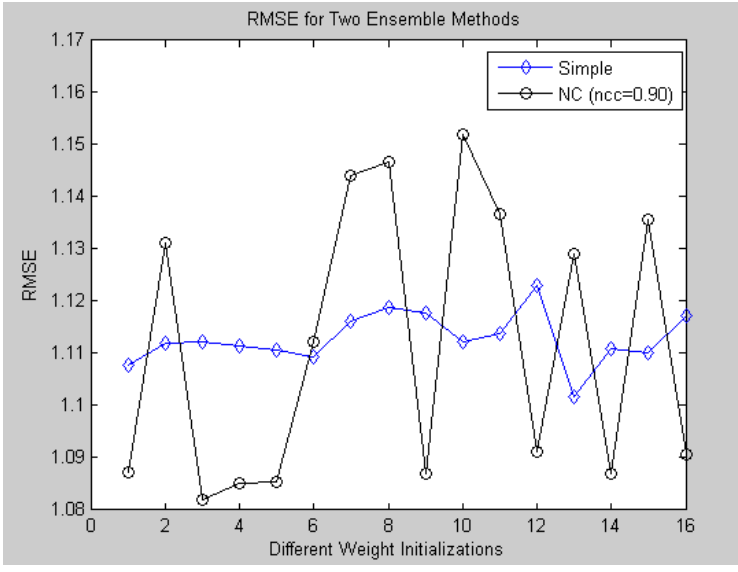

FIGURE 1. RMSE of Two Different Ensembles (ncc: 0.90 in second ensemble)

RMSE (Root Mean Square Error) (101 Epochs)		
Randomization Seed	Simple Ensemble (Individual training of three networks) lr: 0.05 mc:0.9	NC Ensemble (Simultaneous training of three networks) lr: 0.05 mc:0.9 ncc:0.99
1	1.107711434	1.087565718
5	1.111880434	1.089961843
10	1.112127800	1.091227609
15	1.111352544	1.095420215
20	1.110780784	1.085610372
50	1.109321873	1.088455441
100	1.116207508	1.096268318
200	1.118746130	1.092198979
300	1.117772735	1.096931064
400	1.112187941	1.092623514
500	1.113877294	1.098252470
600	1.122937664	1.087077594
700	1.101775745	1.091983110
800	1.110991583	1.091900846
900	1.110193667	1.085509249
1000	1.117178851	1.091349032

Table 2. RMSE of Two Different Ensembles (ncc: 0.99 in second ensemble)

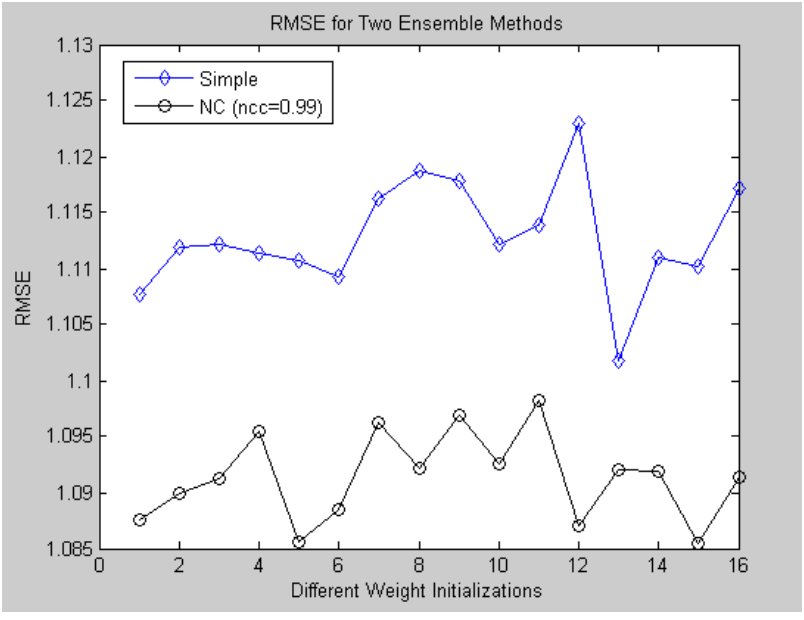

FIGURE 2. RMSE of Two Different Ensembles (ncc: 0.99 in second ensemble)

The results show that the ensemble trained with negative correlation learning performs better than the simple ensemble in which each network in the ensemble trained individually. It can be seen that there is a significant difference when changing the value of negative correlation constant from 0.90 to 0.99. When it was 0.90, almost half of the executions the 'Simple' ensemble performs better. If the correlation penalty strength is increased to 0.99, in the all of the runs the 'NC' ensemble performs better.

The quality of our input data may not have been good enough for the collaborative filtering. As we mentioned earlier, we eliminated, in other words, we did not use some features from our dataset. The result of 1,091395961 (RMSE average of 16 different initializations of the networks in the ensemble) is not good. But negative correlation learning improves the RMSE. When negative correlation learning is not used, RMSE is 1,112815249 (average of 16 different initializations of the networks in the ensemble).

REFERENCES

[1] Herlocker J. L., Konstan J.A., Terveen L.G., Riedl J. T., Evaluating Collaborative Filtering Recommender Systems, A*CM Transactions on Information Systems*, Vol. 22. No. 1. January 2004.

[2] Pryor M. H., The effects of Singular Value Decomposition on Collaborative Filtering, *Computer science Technical Report*, Dartmouth College, PCS-TR98-338, June 1998

[3] Linden G., Smith B., York J., Amazon.com Recommendations Item-to-Item Collaborative Filtering, *IEEE Internet Computing*, January-February 2003.

[4] Bell R. M., Koren Y., Improved Neighborhood-Based Collaborative Filtering, *KDDCup'07* August 2007.

[5] Bell R. M., Koren Y., Volinsky C., Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems, *KDDCup'07* August 2007.

[6] Töscher A., Jahrer M., Legenstein R., Improved Neighborhood-Based Algorithms for Large-Scale Recommender Systems, 2nd *Netflix-KDD Workshop,* August 2008.

[7] Wu M., Collaborative Filtering via Ensembles of Matrix Factorizations, *KDDCup 07,* August 2007.

[8] Bell R. M., Koren Y., Volinsky C., Solution to the Netflix Prize, *The BellKor 2008*

[9] Su X., Khoshgoftaar T. M., Greiner R., Imputation-Boosted Collaborative Filtering Using Machine Learning Classifiers, *SAC 08*, March 2008.

[10] Koren Y., Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering, *KDD 08*, August 2008.

[11] Kuncheva L. I., Skurichina M., Duin R.P.W., An Experimental Study on Diversity for Bagging and Boosting with Linear Classifiers, *Information Fusion 3* (2002).

[12] Yang J., Li K. F., Zhang D., Recommendation Based on Rational Inferences in collaborative filtering, *Knowledge-Based Systems* 22 (2009).

[13] Sharkey A. J. C., Types of Multinet system, *MCS 2002, LNCS 2364*, 2002.

[14] Brown G., Wyatt j., Harris R., Yao X., Diversity Creation Methods: A Survey and Categorization, *Information Fusion xxx* (2004).

[15] Waterhouse S., MacKay D., Robinson T., Bayesian Methods for Mixtures of Experts, *Neural Information Processing Systems* 8.

[16] Jacobs R. A., Jordan M. I., Barto A. G., The Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks, *COINS Technical Report 90-27*, March 1990.

[17] Dietterich T. G., Ensemble Methods in Machine Learning, *MCS 2000*, *LNCS*, 2000.

[18] Perrone M. P., Cooper L. N., When Networks Disagree: Ensemble Methods for Hybrid Neural Networks, *Neural Networks for Speech and Image Processing*, 1993.

[19] Opitz D., Maclin R., Popular Ensemble Methods: an Empirical Study, *Journal of Artificial Intelligence Research* 11 (1999).

[20] Shi H., Lv Y., An Ensemble Classifier Based on Attribute Selection and

diversity Measure, Fifth International *Conference on fuzzy Systems and Knowledge Discovery* 2008.

[21] Brown G., Yao X., On the Effectiveness of Negative Correlation Learning, *First UK Workshop and Computational Intelligence UKCI 01*, September 2001.

[22] Zanda M., Brown G., Fumera G., Roli F., Ensemble Learning in Linearly Combined Classifiers via Negative Correlation, *MCS 2007*, *LNCS 4472*, 2007.

[23] Liu Y., Yao X., Ensemble Learning via Negative Correlation, *Neural Networks* 12 (1999).

[24] Liu Y., Yao X., Higuchi T., Evolutionary Ensembles with Negative Correlation Learning, *IEEE Transactions on Evolutionary Computation*, November 2000.

[25] Ar Y., Bostanci E., A genetic algorithm solution to the collaborative filtering problem, *Expert Systems with Applications*, November 2016.

Current Address: YILMAZ AR: Ankara University, Computer Engineering Department, Ankara TURKEY
E-mail: ar@ankara.edu.tr
ORCID: https://orcid.org/xxxx-xxxx-xxxx-xxxx