American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Performance Comparison of Collaborative Filtering Prediction Methods on Recommendation System

Win Win Moe[a]*, Nang Aye Aye Htwe[b]

[a,b]*Department of Computer Engineering and Information Technology, Mandalay Technological University, Mandalay, Myanmar*

[a]*Email: win2moe.it @gmail.com*

[b]*Email: htwe.aye@gmail.com*

Abstract

Recommendation systems were introduced as the computer-based intelligent techniques to deal with the problem of information overload. Collaborative filtering is a simple recommendation algorithm that executes the similarity (neighborhoods) between items and then computes the missing data predictions. A serious limitation of collaborative filtering is the sparisity problem, referring to the situation where insufficient rating history is available for inferring reliable similarities. This research compares four prediction methods: Weighted Sum, Mean-Centering, Boosted Weighted Sum and Boosted Double Means Centering predictions. Boosting double means centering taken into account information of both users and items in order to overcome the potential decrease of accuracy due to sparsity when predicting the missing value. It tries to capture the user and item biases from the whole effects so as to enable the better concentrating on user-item interaction. Furthermore, ensemble learning will improve the performance collaborative filtering method because an ensemble of collaborative filtering models based on a single collaborative filtering algorithm considered the problem of sparsity, recommender error rate and sample weight update. Rating history in Book-Crossing dataset with 91% sparsity level is used to evaluate the missing rating predictions and the performance comparison of rating predictions on two traditional collaborative filtering and two boosting collaborative filtering frameworks. Experimental results shows that the proposed boosted double mean centering framework improve the prediction accuracy than the two traditional collaborative filtering and the other boosting prediction algorithm.

Keywords: Collaborative filtering; Ensemble learning; Prediction algorithm; Recommendation system; Similarity measure.

----

* Corresponding author.

1. Introduction

Recommendation Systems (RS) are now popular both in commercially and in the research community, where many approaches have been suggested for providing recommendations [1]. RS can now be found in many applications that expose the user to a huge collection of items. Such systems typically provide the user with a list of recommended items that are preferred, or predict how much item is preferred. To predict the unknown rating, collaborative filtering is the most successful and popular method for providing predictions over user preferences. Memory based collaborative filtering mainly deals with user-to-user or item-to-item similarity computation meaning that that system utilizes neighborhoods constructed from a collection of the rating history. In other words, this deals with the overall user-item rating matrix to generate the prediction result. In the real world recommendation, it is usual that the most active users have rated a very limited percentage of items, when compared to the available total. That leads to sparse user-item matrices, inability to locate successful neighbors and finally, the generation of weak recommendations [2]. The proposed framework composed of two main components to overcome the sparsity problem. Firstly, the double mean centering collaborative filtering is proposed that tries to capture user biases and item biases and separates them from the whole effects so as to better concentrating on user-item interaction. Most improvement of collaborative filtering models either create more sophisticated models or add new enhancements to known ones. Ensemble is a machine learning approach that uses a combination of identical models in order to improve the results obtained by a single model. Unlike hybridization methods in recommender systems that combine different types of recommendation models (e.g. a CF model and a content based model), the base model which construct the ensemble is a single learning algorithm. Therefore, the second idea is to propose the boosting framework [3] that will iteratively trains the collaborative filtering prediction. After each predictor is trained, its error rate is measured, and mispredicted instances are emphasized, and then train a new predictor. At the final prediction time, the boosting framework combines the results from the individual predictors trained to produce the final results that will improve the prediction accuracy of traditional collaborative filtering.

The rest of the paper is organized as follows: Section II covers Related Work. Section III presents Recommendation System including four prediction approaches. Section IV provides Experimental results of different recommendation algorithms. Section V concludes the proposed system.

2. Related Work

Bedroll Sarwar and his colleagues [4] presented that the recommender systems can be applied in knowledge discovery techniques to the problem of making personalized recommendations for information or products. These systems find the neighbors using co-rated items and then predict the missing ratings. The bottleneck in conventional collaborative filtering algorithms is the search for neighbours among a large user population of potential neighbours. Item-based algorithms avoid this bottleneck by exploring the relationships between items first, rather than the relationships between users. Different techniques for computing item-item similarities are presented (e.g., item-item correlation and cosine similarities between item vectors) and different techniques for rating prediction (e.g., weighted sum and regression model). Because the relationships between items are relatively static, Item-based algorithm may be able to provide the same quality as the user-based algorithms with less computation time.

Emmanouil Vozalis [5] provided a brief review of various collaborative filtering algorithms that mainly deal with similarity computation constructed as a collection of neighbours. Any recommender system has the two basic entities (the user and the item) and the system operates over the entire user-item matrix to make prediction. It is usual in e-business that even the most active users rated a very limited percentage of products, when compared to the available total. As a result, the user-item matrix is usually sparse, including numerous "no rating" values, thus making it harder for filtering algorithms to generate satisfactory results. Many techniques such as default voting, pre-processing using average and dimensionality reduction have been proposed to improve on the results of the recommendation process. These methods may suffer overfitting and lose of valuable information.

Xiaotian Jiang and and his colleagues [6] discussed Boosting framework that combines multiple homogeneous recommenders, which are based on the same collaborative filtering algorithm with different sample weights. Boosting is employed to enhance the performance of classification by integrating multiple weak classifiers into a better classifier with high accuracy. AdaBoost employs sampling with replacement to generate the actual training dataset for each classifier. In recommender systems, whose dataset is often very sparse, that does not work. Sampling with replacement brings about massive repetitive samples making training dataset even much sparser than the original input dataset. And the reweighting strategy is directly used to control the new classifier how to appropriately update sample weights in its shaping or prediction phase. The boosting framework can improve the performance of memory based collaborative filtering although it suffers from the complexity brought by iteratively calculating and storing the similarity matrix.

3. Recommendation System

Recommendation System is a research carried out in the field of Information Retrieval, particularly into Information Filtering techniques developed for better cope with the exponential growth of information in the computer age. The different types of Recommendation Systems are currently developed, namely Collaborative Filtering, Content Based approach and Hybrid Recommendation System. Among these three methods, the most widely used recommendation algorithm is Collaborative Filtering (CF) that can recommend objects without requirements on contents of items' representation. The main works of collaborative filtering are finding the nearest neighbors that are the similarity computation between items and estimating unknown ratings using the high similarity neighbors with the target users to provide the personalized recommendation list to users.

The input to a recommendation system depends on the type of the employed filtering algorithm. Generally, the input belongs to one of the categories such as ratings, demographic data and content data. Collaborative Filtering Recommender Systems rely on user preferences (ratings) that are expressed by users to items. Formally in a Book Recommender System, ratings are normally provided by the user and follow a specified numerical scale (example: 1-bad to 5-excellent) at the intersection of a user to a book represents that the user votes to that book as shown in figure 1. The value '0' in this figure represents that the user votes to any book and also known as missing rating value. In this research, predicting missing rating with different four collaborative filtering prediction methods are presented.

BOOK RECOMMANDER SYSTEM

Home | Register | Book Gallery | Recommandation | RMSE | Add Books | Manager

ORIGINAL MATRIX FOR WEIGHTED SUM CF

Book/User	Admin	Win Win Moe	Nan Mo Kham	Zun May Myint	May Thu Lwin	Nwe Nwe Win	Daw May Myat Mon	Dr. Myat Su Mon	Aye Aye Myat	Daw May Htet	Daw Su Mon Thu	Daw Nwe Nwe Win	Dr. Phyu Thin	Dr. Thu Thu Win	Daw Thin Thu Tun	Dr. Zin May Zaw	Daw San San Maw	Da Kh Sw Li
B0001	0.00	0.00	0.00	5.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.(
B0002	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.(
B0003	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.(
B0004	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	5.00	5.00	0.00	0.00	0.00	0.(
B0005	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.(
B0006	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	0.00	5.00	4.00	0.00	5.00	4.00	0.00	0.00	0.(
B0007	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.(
B0008	5.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.(
B0009	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.(
B0010	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.(
B0011	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	0.(
B0012	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	4.00	0.00	0.00	0.00	5.00	0.(

Figure 1: original rating matrix

Similarity computation between items or users is a critical step in collaborative filtering based recommendation system. For item-based CF algorithm, the basic idea of the similarity computation between item ***i*** and item ***j*** is first to work on the users who have rated both of these items. Similarity computation is applied to determine the similarity, s_{ij}, between the two co-rated items of the users. Various similarities have been suggested, namely Euclidean Distance, Cosine Distance, Pearson Correlation Coefficient and others. Among them, Pearson Correlation Coefficient was suggested as a suitable measure in Collaborative Filtering. Pearson correlation coefficient measures the extent to which two variables linearly relate with each other as in equation (1). For the item-based algorithm, the Pearson correlation between item ***i*** and item ***j*** is

$$s_{ij} = \frac{\sum_{j \in N(i)} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{j \in N(i)} (r_{ui} - \bar{r}_i)^2} \sqrt{\sum_{j \in N(i)} (r_{uj} - \bar{r}_j)^2}} \qquad (1)$$

Home | Register | Book Gallery | Recommandation | RMSE | Add Books | Manager

BOOK TO BOOK SIMILARITY FOR WEIGHTED SUM CF

Book/Book	B0001	B0002	B0003	B0004	B0005	B0006	B0007	B0008	B0009	B0010	B0011	B0012	B0013	B0014	B0015	B0
B0001	1.00	0.97	1.00	1.00	-1.00	-1.00	-1.00	0.00	0.93	0.00	0.20	-1.00	1.00	-1.00	0.20	0
B0002	0.97	1.00	-0.41	-1.00	-0.20	-0.62	0.00	0.00	-0.39	0.00	-1.00	1.00	0.00	0.03	1.00	-0
B0003	1.00	-0.41	1.00	0.00	0.10	1.00	-1.00	-1.00	-1.00	0.00	0.00	-0.51	0.00	-0.08	0.00	1
B0004	1.00	-1.00	0.00	1.00	0.00	0.90	0.45	0.00	1.00	0.00	0.78	-0.21	0.00	-0.12	0.30	0
B0005	-1.00	-0.39	0.10	0.00	1.00	-1.00	0.00	0.00	0.00	0.00	-1.00	0.39	0.39	1.00	-1.00	-1
B0006	-1.00	-0.62	1.00	0.90	-1.00	1.00	-1.00	-1.00	-0.20	0.00	-0.20	0.27	-0.49	0.16	-0.36	0
B0007	-1.00	0.00	-1.00	0.45	0.00	-1.00	1.00	1.00	0.63	-0.71	0.55	0.66	1.00	0.47	0.00	0
B0008	0.00	0.00	-1.00	0.00	0.00	-1.00	1.00	1.00	0.82	0.00	0.97	1.00	0.42	0.39	1.00	1
B0009	0.93	-0.39	-1.00	1.00	0.00	-0.20	0.63	0.82	1.00	0.00	0.57	1.00	1.00	-0.48	0.98	0
B0010	0.00	0.00	0.00	0.00	0.00	0.00	-0.71	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0
B0011	0.20	-1.00	0.00	0.78	-1.00	-0.20	0.55	0.97	0.57	0.00	1.00	-0.47	0.99	-0.02	0.69	0
B0012	-1.00	1.00	-0.51	-0.21	0.39	0.27	0.66	1.00	1.00	0.00	-0.47	1.00	0.44	1.00	0.94	-0
B0013	1.00	0.00	0.00	0.00	0.39	-0.49	1.00	0.42	1.00	0.00	0.99	0.44	1.00	-0.01	0.58	1
B0014	-1.00	0.03	-0.08	-0.12	1.00	0.16	0.47	0.39	-0.48	0.00	-0.02	1.00	-0.01	1.00	0.12	-0
B0015	0.20	1.00	0.00	0.33	-1.00	-0.36	0.00	1.00	0.98	0.00	0.69	0.94	0.58	0.12	1.00	0
B0016	0.72	-0.80	1.00	0.77	-1.00	0.05	0.04	1.00	0.63	0.00	0.45	-0.25	1.00	-0.55	0.41	1

Figure 2: book-to-book similarity

Using the Pearson Correlation, the similarity is represented in a scale [-1 , +1], where a positive high value suggests a high correlation, a negative high value suggest inversely high correlation, and lastly a zero correlation indicates uncorrelated items shown in figure 2. Sparsity, a result of many users rated only a small portion of many items, hinders the search for similar items, and hampers the preference estimation of unrated items, which in turn make it difficult for the recommender system to output a successful recommendation.

Ensemble has been proposed to improve the result of the recommendation process, which decreased due to sparsity. The main idea of the ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions than those produced by a single model. AdaBoost (Adaptive Boosting), which was first introduced [7], is a popular ensemble algorithm that improves the simple boosting algorithm via an iterative process. The first recommendation algorithm is applied on original rating data set with the uniform distribution of sample weights. In each iteration, the weights of all misclassified instances are increased while the weights of correctly classified instances are decreased. Consequently, the weak learner is forced to focus on the difficult instances of the training set by performing additional iterations and creating more classifiers. According the prediction results in each recommender, an error rated is obtained and this error rate is assigned a weight (this is the predicted error weight not sample weight to obtain the strong recommendation algorithm. The detail process of AdaBoost framework for memory-based collaborative filtering is presented in the following algorithm.

Input: T_{train} - Training dataset of ratings (u,i,r_{ui}); M - the ensemble size;

Base CF - the traditional collaborative filtering prediction algorithm

Train Model

1. Apply Base CF to predict rating with the T_{train}

2. Assign the weight of each sample in T_{train} with a value $m \in [1,M]$ do

3. Train the recommender R_m using boosted weighted collaborative filtering prediction

4. Calculate the error rate of Recommender R_m with error equation

$$err(R_m) = \sum_{(u,i)\in T_{train}} \left(\frac{\left|\hat{r}_{ui} - \hat{r}_{ui}^{m}\right| \times \sum_{j\in B(i;u,k)} w_{uj}^{m}}{\sum_{(u,i)\in T_{train}} \sum_{j\in B(i;u,k)} (w_{uj}^{m} \cdot \left|\hat{r}_{ui} - \hat{r}_{ui}^{m}\right|)} \times \frac{\left|\hat{r}_{ui} - \hat{r}_{ui}^{m}\right|}{D} \right)$$

5. Adjust the sample weight with weight update formula for each of its k-nearest neighbors

$$w_{uj}^{m+1} = w_{uj}^{1} \times (1 + sgn_{ui}^{m}(j) \times \frac{err(R_m)}{1 - err(R_m)} \times UE_{ui}^{m} \times \rho)$$

6. Normalize these sample weights end for

Output: The final predicted rating for the model is produced by averaging

$$p_{ui} = \sum_{i=1}^{M} \hat{r}_{ui}^{m} / M$$

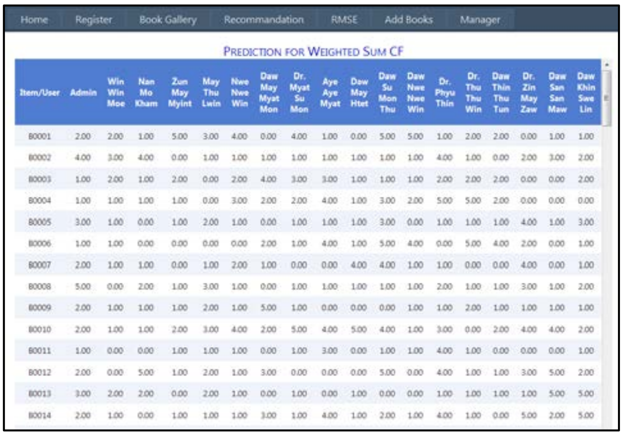
Algorithm 1: AdaBoost framework for memory-based collaborative filtering

3.1. Traditional Collaborative Filtering Predictions

To obtain the predictions and recommendations is the final important step in collaborative filtering algorithm. The two traditional CF predictions: Weighted Sum prediction and Mean-Centering prediction are presented in this section. In a CF algorithm, after calculating the similarity, a subset of nearest neighbors of the active item is chosen based on items' similarity values, and a weighted aggregate of the ratings is used to generate predictions for the active user. In order to produce an estimation of a rating for an unrated item, a weighted average of all available ratings is done, with the correlation values of the neighborhoods as shown in equation (2).

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u(i)} s_{ij} r_{uj}}{\sum_{j \in N_u(i)} \left| s_{ij} \right|} \tag{2}$$

,where $N_u(i)$ represents the number of neighbors that have user u in common with book i, of which j is a particular neighbor book, and s_{ij} is the similarity between the book i and one of its neighbors' j. The prediction results using traditional weighted sum prediction is shown in figure 3. Since this method is only depends on the neighborhood correlation, the predicted values are very sensitive.



Item/User	Admin	Win Win Moe	Nan Mo Kham	Zun May Myint	May Thu Lwin	Nwe Nwe Win	Daw May Myat Mon	Dr. Myat Su Mon	Aye Aye Myat	Daw May Htet	Daw Su Mon Thu	Daw Nwe Nwe Win	Dr. Phyu Thin	Dr. Thu Thu Win	Daw Thin Thu Tun	Dr. Zin May Zaw	Daw San San Maw	Daw Khin Swe Lin
B0001	2.00	2.00	1.00	5.00	3.00	4.00	0.00	4.00	1.00	0.00	5.00	5.00	1.00	2.00	2.00	0.00	1.00	1.00
B0002	4.00	3.00	4.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	4.00	1.00	0.00	2.00	3.00	2.00
B0003	1.00	2.00	1.00	2.00	0.00	2.00	4.00	3.00	3.00	1.00	1.00	1.00	2.00	2.00	2.00	0.00	0.00	2.00
B0004	1.00	1.00	1.00	1.00	0.00	3.00	2.00	2.00	4.00	1.00	3.00	2.00	5.00	5.00	2.00	0.00	0.00	0.00
B0005	3.00	1.00	0.00	1.00	2.00	1.00	0.00	1.00	1.00	1.00	3.00	0.00	1.00	1.00	1.00	4.00	1.00	3.00
B0006	1.00	1.00	0.00	0.00	0.00	0.00	2.00	1.00	4.00	1.00	5.00	4.00	0.00	5.00	4.00	2.00	0.00	1.00
B0007	2.00	1.00	1.00	0.00	1.00	2.00	1.00	0.00	0.00	4.00	4.00	1.00	1.00	0.00	0.00	4.00	0.00	1.00
B0008	5.00	0.00	2.00	1.00	3.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	1.00	3.00	1.00	2.00
B0009	2.00	1.00	1.00	1.00	2.00	1.00	5.00	1.00	0.00	0.00	0.00	1.00	1.00	2.00	1.00	1.00	1.00	1.00
B0010	2.00	1.00	1.00	2.00	3.00	4.00	2.00	5.00	4.00	5.00	4.00	1.00	3.00	0.00	2.00	4.00	4.00	2.00
B0011	1.00	0.00	0.00	1.00	1.00	1.00	0.00	1.00	3.00	0.00	1.00	1.00	4.00	1.00	0.00	0.00	0.00	1.00
B0012	2.00	0.00	5.00	1.00	2.00	1.00	3.00	0.00	0.00	0.00	5.00	0.00	4.00	1.00	1.00	3.00	5.00	2.00
B0013	3.00	2.00	2.00	0.00	2.00	1.00	0.00	1.00	0.00	1.00	0.00	0.00	1.00	1.00	1.00	1.00	5.00	5.00
B0014	2.00	1.00	0.00	1.00	1.00	1.00	3.00	1.00	4.00	1.00	2.00	1.00	4.00	1.00	0.00	5.00	2.00	5.00

Figure 3: Predicted ratings using weighted-sum collaborative filtering

One of the problems faced with using ratings as a means of representing taste, is that each user has personal interpretation of the scale. While one rater might tend to give high marks to item, another rater might keep the highest grades for exceptional items. The second method, *Mean Centering* approach re-maps a user's rating by subtracting the mean value of all that user ratings, effectively signaling if the particular rating is positive or negative when compared to the mean. Positive values represent above-average ratings, negative values represent below-average ratings, and zero represents an average rating. Gives the formulation (3):

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in N_u(i)} s_{ij}(r_{uj} - \bar{r}_j)}{\sum_{j \in N_u(i)} |s_{ij}|} \quad (3)$$

where $\bar{r}_i$ and $\bar{r}_j$ are the average ratings for the item i and item j on all other rated items and the similarity between item i and item j is s_{ij}. $\hat{r}_{ui}$ is the predicted rating of mean centering prediction and these results are shown in figure 4, where the input dataset is often suffered from item bias resulting the mean-centering oftentimes predicts values outside the rating scale.

Home | Register | Book Gallery | Recommandation | RMSE | Add Books | Manager

PREDICTION FOR MEAN CENTERING CF

Item/User	Admin	Win Win Moe	Nan Mo Kham	Zun May Myint	May Thu Lwin	Nwe Nwe Win	Daw May Myat Mon	Dr. Myat Su Mon	Aye Aye Myat	Daw May Htet	Daw Su Mon Thu	Daw Nwe Nwe Win	Dr. Phyu Thin	Dr. Thu Thu Win	Daw Thin Thu Tun	Dr. Zin May Zaw	Daw San San Maw	Daw Khin Swe Lin
B0001	4.00	4.00	4.00	5.00	4.00	4.00	4.00	4.00	3.00	4.00	3.00	5.00	4.00	4.00	4.00	4.00	4.00	4.00
B0002	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
B0003	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
B0004	3.00	3.00	4.00	3.00	3.00	4.00	3.00	4.00	4.00	4.00	4.00	4.00	5.00	5.00	4.00	3.00	3.00	3.00
B0005	4.00	4.00	4.00	4.00	3.00	4.00	4.00	4.00	3.00	4.00	3.00	4.00	3.00	3.00	3.00	4.00	3.00	4.00
B0006	3.00	3.00	4.00	4.00	4.00	3.00	4.00	4.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	3.00	4.00	3.00
B0007	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
B0008	5.00	4.00	4.00	4.00	3.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
B0009	4.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
B0010	6.00	5.00	6.00	5.00	5.00	5.00	6.00	5.00	5.00	5.00	6.00	6.00	5.00	6.00	6.00	5.00	6.00	5.00
B0011	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	3.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
B0012	4.00	4.00	5.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00
B0013	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	5.00	5.00
B0014	4.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00	5.00

Figure 4: Predicted ratings using mean-centering collaborative filtering

3.2. Boosting Collaborative Filtering Predictions

In order to overcome the sparsity problem, boosting weighted sum and boosting double mean centering prediction are presented by applying the AdaBoost framework. Firstly the traditional weighted sum is boosted, the prediction formula is in turn depending on the sample weights as shown in equation (4). The sample weights for all nearest neighbours w_{uj}^m are initialized uniformly with a value.

$$\hat{r}_{ui}^m = \frac{\sum_{j \in B(i;u,k)} |s_{ij}| \cdot w_{uj}^m \cdot r_{uj}}{\sum_{j \in B(i;u,k)} |s_{ij}| \cdot |w_{uj}^m|} \quad (4)$$

where, $\hat{r}_{ui}^m$ denotes the predicted rating of the i^{th} item by the u^{th} user in the m^{th} recommender. In each iteration the error rate for each recommender, signal function and user error for each neighbours are executed to adjust the neighbourhood samples for next round. If the base two-class classifier used in AdaBoost framework has an recommender error rate of just slightly less than 0.5, the training error of the classifier will approach to zero. For problem with more than two classes (such as numerical preference prediction), the recommender error rate that is less than 0.5 is harder to achieve. This method iteratively predicted the missing values that predicted results are less sensitive than the traditional weighted sum prediction and shown in figure 5.

Home | Register | Book Gallery | Recommandation | RMSE | Add Books | Manager

FINAL PREDICTION FOR BOOST WEIGHTED SUM CF

Item/User	Admin	Win Win Moe	Nan Mo Kham	Zun May Myint	May Thu Lwin	Nwe Nwe Win	Daw May Myat Mon	Dr. Myat Su Mon	Aye Aye Myat	Daw May Htet	Daw Su Mon Thu	Daw Nwe Nwe Win	Dr. Phyu Thin	Dr. Thu Thu Win	Daw Thin Thu Tun	Dr. Zin May Zaw	Daw San San Maw	Da Khi Sw Li
80001	5.00	4.00	5.00	5.00	5.00	4.00	5.00	4.00	4.00	4.00	5.00	5.00	4.00	5.00	4.00	5.00	5.00	5.0
80002	5.00	4.00	4.00	4.00	5.00	4.00	5.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	5.00	5.00	4.0
80003	5.00	4.00	4.00	4.00	5.00	4.00	5.00	5.00	4.00	4.00	5.00	4.00	5.00	5.00	5.00	4.00	4.00	5.0
80004	5.00	4.00	4.00	4.00	5.00	5.00	5.00	5.00	4.00	4.00	5.00	5.00	5.00	5.00	4.00	5.00	4.00	4.0
80005	5.00	4.00	4.00	5.00	5.00	4.00	0.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	5.00	4.00	5.0
80006	5.00	4.00	4.00	4.00	5.00	4.00	5.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	4.00	4.00	4.0
80007	5.00	4.00	5.00	5.00	4.00	5.00	5.00	5.00	4.00	4.00	4.00	5.00	4.00	5.00	4.00	4.00	4.00	4.0
80008	5.00	4.00	4.00	4.00	3.00	4.00	5.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	4.00	4.00	4.0
80009	5.00	5.00	4.00	4.00	4.00	4.00	5.00	5.00	4.00	4.00	5.00	5.00	4.00	5.00	4.00	5.00	5.00	4.0
80010	0.00	5.00	0.00	4.00	5.00	5.00	5.00	5.00	4.00	5.00	4.00	0.00	4.00	0.00	4.00	4.00	4.00	4.0
80011	5.00	4.00	4.00	4.00	5.00	4.00	5.00	5.00	3.00	4.00	5.00	4.00	4.00	5.00	4.00	5.00	4.00	4.0
80012	5.00	4.00	5.00	5.00	5.00	5.00	5.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	5.00	5.00	5.00	4.0
80013	5.00	4.00	4.00	4.00	5.00	4.00	5.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	5.0
80014	5.00	4.00	4.00	4.00	5.00	4.00	5.00	5.00	4.00	4.00	5.00	5.00	4.00	5.00	4.00	5.00	5.00	5.0

Figure 5: Predicted ratings using boosted weighted sum collaborative filtering

Collaborative Filtering focuses on modeling the intersection between users and items. However, many effects that contribute to the ratings, such as user bias and item bias, are not associated with this interaction. User bias indicates the observed difference among users, while item bias indicates the observed bias for each item compared to the overall average. For example, a lower user bias corresponds to a critical user who is prone to rate low, and a high item bias corresponds to a good item that earns higher ratings. Double means centering *avg(u,i)* tries to capture these biases and separate them from the whole effects so as to enable other models better concentrating on user-item interaction. By composing of two components: user bias and item bias, the prediction formula is turned into the form

$$\hat{r}_{ui}^{m} = avg(u,i) + \frac{\sum_{j \in B(i;u,k)} \left| s_{ij} \right| \cdot w_{uj}^{m} \cdot (r_{uj} - avg(u,i))}{\sum_{j \in B(i;u,k)} \left| s_{ij} \right| \cdot \left| w_{uj}^{m} \right|} \tag{5}$$

$$avg(u,i) = \frac{\sum_{\forall i} r_{ui} / \left| r_{u*} \right| + \sum_{\forall u} r_{ui} / \left| r_{i*} \right|}{2} \tag{6}$$

Home | Register | Book Gallery | Recommandation | RMSE | Add Books | Manager

FINAL PREDICTION FOR BOOST DOUBLE MEAN CF

Item/User	Admin	Win Win Moe	Nan Mo Kham	Zun May Myint	May Thu Lwin	Nwe Nwe Win	Daw May Myat Mon	Dr. Myat Su Mon	Aye Aye Myat	Daw May Htet	Daw Su Mon Thu	Daw Nwe Nwe Win	Dr. Phyu Thin	Dr. Thu Thu Win	Daw Thin Thu Tun	Dr. Zin May Zaw	Daw San San Maw	Da Khi Sw Li
80001	4.00	4.00	4.00	5.00	4.00	5.00	5.00	4.00	4.00	4.00	5.00	5.00	4.00	4.00	4.00	5.00	4.00	5.0
80002	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	4.00	4.00	5.00	5.00	4.00	4.00	4.00	5.00	5.00	4.0
80003	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00	5.00	5.00	5.00	5.00	4.00	5.0
80004	5.00	4.00	4.00	4.00	4.00	5.00	4.00	5.00	4.00	4.00	4.00	4.00	5.00	5.00	5.00	5.00	4.00	4.0
80005	5.00	4.00	4.00	3.00	4.00	4.00	0.00	5.00	4.00	4.00	3.00	4.00	4.00	4.00	4.00	5.00	5.00	5.0
80006	5.00	4.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	5.00	4.00	4.0
80007	4.00	4.00	5.00	4.00	4.00	4.00	5.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.0
80008	5.00	4.00	4.00	5.00	3.00	4.00	5.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.0
80009	4.00	4.00	5.00	5.00	4.00	5.00	5.00	4.00	4.00	4.00	5.00	5.00	5.00	5.00	5.00	4.00	5.00	4.0
80010	0.00	3.00	0.00	4.00	3.00	4.00	5.00	4.00	4.00	5.00	5.00	0.00	3.00	0.00	4.00	5.00	4.00	5.0
80011	4.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00	3.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.0
80012	5.00	5.00	5.00	4.00	5.00	5.00	5.00	5.00	4.00	4.00	5.00	4.00	4.00	5.00	4.00	5.00	5.00	5.0
80013	4.00	5.00	4.00	5.00	5.00	5.00	5.00	4.00	4.00	4.00	4.00	4.00	4.00	5.00	4.00	4.00	5.00	5.0
80014	5.00	4.00	5.00	4.00	5.00	4.00	5.00	5.00	4.00	4.00	4.00	5.00	4.00	4.00	4.00	5.00	5.00	5.0

Figure 6: Predicted ratings using boosted double means centering collaborative filtering

The prediction results using Boosted Double Means Centering are shown in figure 6. The major considerations for the adaptation process include sparsity, recommendation error rate and sample weight update. After predicting the missing rating values, it needs to focus mispredicted values of the previous round. This error rate affects how the sample weights of the neighborhood are updated. In addition, η denotes how much the average sample error influences the update process. In this research, the proposed system sets η=0.5 by experience. This system increases the sample weight that is not correctly learned by the previous predictor or decreases sample weight that is correctly learned by the previous round. The resulted normalized sample weights in one of the steps in boosted double mean centering collaborative filtering framework are shown in figure 7.

Sample weights

Item/User	Admin	Win Win Moe	Nan Mo Kham	Zun May Myint	May Thu Lwin	Nwe Nwe Win	Daw May Myat Mon	Dr. Myat Su Mon	Aye Aye Myat	Daw May Htet	Daw Su Mon Thu	Daw Nwe Nwe Win	Dr. Phyu Thin	Dr. Thu Thu Win	Daw Thin Thu Tun	Dr. Zin May Zaw	Daw San San Maw	Daw Khin Swe Lin
B0001	0.74	0.19	1.00	1.00	0.19	1.00	0.73	0.75	0.19	0.19	0.86	1.00	0.60	0.73	0.67	0.71	0.68	0.67
B0002	0.74	0.19	1.00	0.63	0.19	1.00	0.73	0.72	0.19	0.19	0.86	0.19	0.60	0.73	0.67	0.71	0.90	0.67
B0003	0.74	0.19	1.00	0.63	0.19	1.00	0.81	0.72	0.19	0.19	0.71	0.19	0.60	0.73	0.67	0.71	0.68	0.67
B0004	0.74	0.19	1.00	0.63	0.19	1.00	0.73	0.72	0.19	0.19	0.71	0.19	1.00	0.73	0.67	0.71	0.68	0.67
B0005	0.74	0.19	1.00	0.63	0.19	1.00	0.81	0.72	0.19	0.19	0.86	0.19	0.60	0.73	0.67	0.71	0.91	0.67
B0006	0.74	0.19	1.00	0.63	0.19	1.00	0.73	0.72	0.19	0.19	0.71	0.19	0.60	0.73	0.67	0.75	0.91	0.67
B0007	0.74	0.19	1.00	0.63	0.19	1.00	0.73	0.72	0.19	0.19	0.86	0.19	0.60	0.73	0.67	0.75	0.91	0.67
B0008	0.74	0.19	1.00	0.63	0.61	1.00	0.73	0.72	0.19	0.19	0.86	0.19	0.60	0.73	0.67	0.75	0.91	0.67
B0009	0.74	0.19	1.00	0.63	0.19	0.68	0.73	0.72	0.19	0.19	0.71	0.19	0.60	0.73	0.67	0.71	0.68	0.67
B0010	0.75	0.19	1.00	0.63	0.19	1.00	0.73	0.75	0.19	1.00	0.86	1.00	0.60	0.80	0.67	0.75	0.91	0.67
B0011	0.75	0.19	1.00	0.63	0.19	1.00	0.81	0.75	0.59	0.19	0.86	0.19	0.60	0.80	0.67	0.75	0.91	0.67
B0012	0.74	1.00	1.00	0.63	0.19	0.68	0.73	0.72	0.19	0.19	0.71	0.19	0.60	0.73	1.00	0.71	0.68	1.00
B0013	0.74	0.19	1.00	0.63	0.19	1.00	0.73	0.72	0.19	0.19	0.71	0.19	0.60	0.73	0.67	0.71	0.68	1.00
B0014	0.74	0.19	1.00	0.63	0.19	0.68	0.73	0.72	0.19	0.19	0.71	0.19	0.60	0.73	0.67	0.71	0.68	1.00
B0015	0.74	1.00	1.00	0.63	0.19	1.00	0.73	0.72	0.19	0.19	0.71	0.19	0.60	0.73	0.67	0.71	0.68	0.67

Figure 7: normalized neighbors' sample weights

The Adaboost framework uses the same training dataset over and over again and keeps these predicted reults. At the final step, all the predicted values from each round are combined by averaging previously presented in AdaBoost framework and produced the top-N recommendation in a ranked list to active user. The Top-10 book recommendation list produced by Boosting Double Means Centering Prediction is shown in figure 8. When a user log into that user's account and rated at least two items, the system will recommend a list of books that may be the interest of that user.

Figure 8: top-10 recommendation list for active user recommended by boosted double-mean centering collaborative filtering

4. Experimental Results

To evaluate the recommendation accuracy of two traditional collaborative filtering and two Boosting frameworks, Root Mean Square Error (RMSE) is used on rating history in Book Crossing Dataset with sparsity level of 91%. RMSE measures the deviation of predictions generated by the recommendation system from the true ratings values as shown in equation (7).

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in T_{test}} (r_{ui} - \hat{r}_{ui})^2}{|T_{test}|}} \quad (7)$$

Furthermore, total coverage is computed as the fraction of items for which a prediction was generated over the total number of items that all available users have rated in the initial user-item matrix as shown in equation (8). A low coverage value indicates that the recommender system will not able to assist the user with many of items that the user has not rated.

$$coverage = \frac{\sum_{i=1}^{m} n\hat{r}_i}{\sum_{i=1}^{m} n_i} \quad (8)$$

Table 1 shows the RMSE and coverage comparison on four predictions. The weighted sum prediction is the worst prediction method with the recommendation error 3.33. However, ensemble framework can reduce the recommendation error of weighted sum to 1.24. Although, the mean centering results are sometimes exceeds the input rating scale, it can improve the recommendation error to 1.18 and gives the highest coverage.

Table 1: Performance Comparison on Two Traditional CF versus Two Boosting CF Frameworks

Prediction Methods	RMSE	Coverage (%)
Weighted Sum Prediction	3.33	51.6%
Mean Centering Prediction	1.18	99.1%
Boosted Weighted Sum Prediction	1.24	93.9%
Boosted Double Mean Centering Prediction	1.11	93.8%

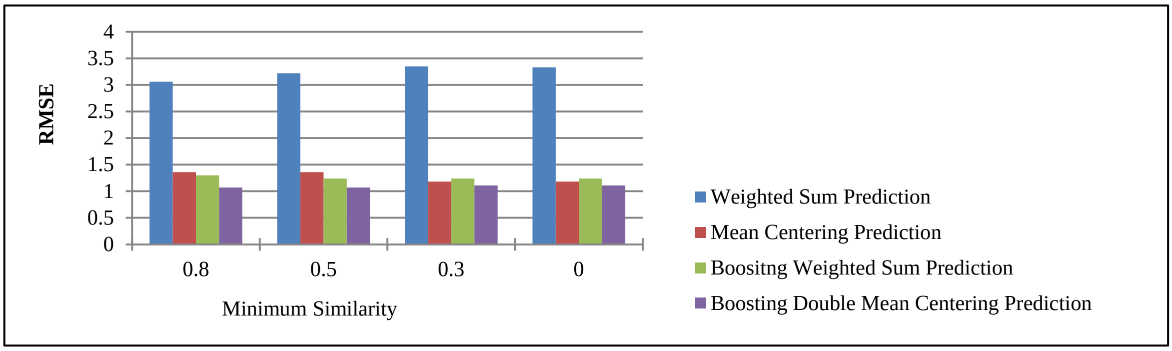


Figure 9: performance comparison in RMSE with different maximum similarity score

The proposed boosted double mean centering approach can predict least recommendation error than the other method although it has slightly low coverage than the mean centering prediction as shown in table 1. Figure 9 shows the RMSE results of the four methods evaluated varying minimum similarity count. The boosted double means centering can predict minimum recommendation error than the other method whatever the amount of nearest neighbors are varied.

5. Conclusion

In this research, four collaborative prediction methods are presented how to improve the recommendation performance that can decrease due to sparsity. The traditional collaborative filtering predicts the missing rating depending on the nearest neighbors and neighbors' ratings. So the traditional weighted sum is sensitive and it can produce highest recommendation error and coverage is least when the input dataset is extremely sparse. Since the mean centering prediction remaps the missing rating by comparing the items mean rating, it slightly improves both error rate and coverage except it suffers from item biases. The weighted sum prediction is boosted by iteratively predicting the missing rating, so it can produce better recommendation than the traditional weighted sum. The boosted double mean centering approach tries to capture both user and item biases and it also reduce recommender error by iteratively predicted with updated sample weight to overcome the sparsity problem. So, the proposed boosted double means centering can increase the prediction accuracy with least RMSE than the other three methods. The boosting framework can improve recommendation accuracy although it can suffer the complexity brought by iteratively calculating and storing the similarity matrix. Furthermore, these systems cannot be recommending a new item unless a user has rated it before due to neighbor formation.

Acknowledgement

I would like to express my deepest gratitude and sincere appreciation to the following persons, whose guidance aided towards the competition of this research. First, I would like to thank Dr. Nang Aye Aye Htwe for her kind permission, patient supervision and precious guidance to conduct this paper. I would like to thank my parents, teachers and all of friends for their support and encouragement throughout this thesis.

References

[1] Greg Linden, Brent Smith, and Jeremy York, “Recommendations Item-to-Item Collaborative Filtering”, IEEE INTERNET COMPUTING, JANUARY-FEBUARY, 2003.

[2] Claudio Adrian Levinas, “An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals”, September 2014.

[3] Alon Schclar, Alexander Tsikinovsky and Lior Rokach, “Ensemble Methods for Improving the Performance of Neighborhood-based Collaborative Filtering”, October 23-25, 2009, New York, USA.

[4] Badrul Sarwar, George Karypis, Joseph Konstan and John Riedl, “Item-Based Collaborative Filtering Recommendation Algorithms”, In Proceedings of the 10th international conference on World Wide Web, pages 285-295. ACM, 2001.

[5] Emmanouil Vozalis and Konstantinos G. Margaritis, “Analysis of Recommender Systems' Algorithms”,

http://macedonia.uom.gr.

[6] Xiaotian Jiang and et al "Novel Boosting Frameworks to Improve the Performance of Collaborative Filtering", JMLR: Workshop and Conference Proceedings 29:87-99, 2013.

[7] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm", Machine Learning: Proceedings of the Thirteenth International Conference, 1996.

[8] R. Sahal, S. Selim and A. Eikorany, "An Adaptive Framework for Enhancing Recommendation using Hybrid Techniques", International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 2, April, 2014.

[9] A. Bellogin and A. P. Vries, "Understanding Similarity Metrics in Neighbour-based Recommender Systems".

[10] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques", Advance in Artificial Intelligence, 2009.

[11] P. Adamopoulos and A. Tuzhilin, "On Over-Specialization and Concentration Bias of Recommendation: Probabilistic Neighborhood Selection in Collaborative Filtering Systems", 2014.

[12] C. Lili, "Recommender Algorithms based on Boosting Ensemble Learning", International Journal on Smart Sensing and Intelligent System Vol 8, No 1, March 2015.