IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 24, No. 2, April, 2019, pp. 131-137
Journal homepage: http://iieta.org/journals/isi

Improving the Accuracy of M-distance Based Nearest Neighbor Recommendation System by Using Ratings Variance

Narges Hasanzadeh, Yahya Forghani*

Islamic Azad University, Mashhad branch, Mashhad, Iran

Corresponding Author Email: yahyafor2000@yahoo.com

https://doi.org/10.18280/isi.240201

Received: 10 January 2019
Accepted: 3 April 2019

Keywords:
M-distance, recommendation system, MBR, collaborative filtering, nearest neighbor

ABSTRACT

M-distance based recommendation system (MBR) is a nearest neighbor based recommendation method which uses the average of ratings given to an item as the attribute of that item. This attribute is used to determine similar items. Then, the average of the rating given to the similar items to an item of the active user determines the rating of that item. In this paper, to decrease the error of MBR, by combining the following ideas, eight MBR-based recommendation systems are proposed: (a) Using the variance of item ratings in addition to the average of item ratings, as two attributes of an item, for determining similar items in an item-based nearest neighbor method; (b) Using the variance of user ratings in addition to the average of user ratings, as two attributes of a user, for determining similar users in a user-based nearest neighbor method; (c) Using a weighted average method for combining the ratings of similar items or similar users; (d) Using ensemble learning. Experimental results on real datasets show that our proposed EVMBR and EWVMBR which use ensemble learning have the least error. The error of the suggested EWVMBR is at-least 20% lower than that of MBR, Slope-One, P-kNN, and C-kNN.

1. INTRODUCTION

In general, recommendation systems have two tasks: prediction and suggestion. In prediction task, the recommendation system, based on the available information such as the history of the comments and ratings of the active user and the other users on some items, tries to predict the rating of the active user on a new item. The recommendation system intends to generate new suggestions for an active user. User's interest in an item is indicated by the user's ratings on that item. The suggestion task involves advising a list of items that are more likely to be close to the target user's tastes. Various methods have been proposed for prediction and suggestion in various articles. The accuracy and runtime are the most important challenges of recommendation systems.

The user-based and item-based collaborative filtering is simple prediction and suggestion techniques which give an acceptable level of accuracy. Sarwar et al. [1] predicted active user ratings on a new item based on the active user's ratings for similar items (or neighboring items), and therefore, this method is called an item-based approach. The different criterion can be used to determine similar items. A new similarity criterion [2] was proposed to increase the accuracy of collaborative filtering methods for a sparse user-item table. This new similarity criterion also solves the cold start problem. The experiments show that this method is more efficient than other similarity criteria such as cosine similarity and Pearson correlation coefficient-based similarity. The reversed collaborative filtering [3] predicts unrated items of a user based on k-nearest neighbors of the rated items. The set of k-nearest neighbor of rated items can be determined in the training phase. This set is usually much smaller than the set of all items. Therefore, the speed of reverse collaborative filtering method is better than that of traditional collaborative filtering methods.

To increase the speed of k-nearest neighbor methods, a new criterion called M-distance was introduced to calculate the similarity of items [4]. The M-distance of two items is the absolute of the difference of the average ratings of those two items. The rating of the active user on an unrated item is set to the average rating of a number of its most similar rated items. The speed of M-distance based method is much better than that of cosine similarity and Pearson correlation based approaches because to determine the similarity of two items, M-distance compares two scalars, i.e. two averages, while in cosine similarity and Pearson correlation coefficient method, two rating vectors, i.e. the ratings of all users on those two items, are compared.

In the Slope-One [5], a simple linear regression model is used to predict ratings. In extended Slope-One [6], first, some similar users to the active user are determined. Then, the Slope-One algorithm, based on just these similar users and not all of the users, estimates the rating of unrated items of the active user. The incremental Slope-One [7] is suitable for the condition where the ratings of users on items are completed incrementally.

Multi-class Co-Clustering (MCoC) is a co-clustering method which groups items and users such that the users with similar interests and their interest items are in the same group [8]. Then, each time, a collaborative filtering method is implemented for a group of users and items.

Contrary to the collaborative filtering which determines the rating of an item based on other users' ratings on that item, in a content-based filtering approach, the rating of the active user on an item is estimated based on the item attributes and the user's interests in those attributes [9]. The advantage of

content-based filtering approach is that to determine the rating of a user on an item, it is not necessary to collect the ratings of different users on that item. Content-based filtering can be used also to strengthen collaborative filtering [10].

In the most of recommendation systems, the prediction of popularity or not the popularity of an item for a user, or item suggestion or not suggestion action, is considered as a two-way decision or a two-class classification problem. In the three-way decision method [11-13], in addition to the two mentioned decisions, i.e. item suggestion and not the suggestion, a third decision may also be made. In other words, if this method is rather quite certain about the popularity of an item for a user, the item is suggested to the user. If the method is unsure, it does not offer it. Otherwise, a third decision is made. This third decision can be taking guidance from the user.

A three-way decision method which uses a random forest method to determine the popularity of an item for a user was proposed [14]. In this method, after defining a specific objective function, two thresholds are determined in such a way that this objective function is minimized. Then, if the level of popularity of an item for a user determined by the random forest is greater than the first threshold, the item is suggested to the user. If the level of popularity of an item for the user is less than the second threshold, the item is not offered to the user. Otherwise, it takes guidance from the user.

In matrix factorization method [15], the latent features of users and items are determined in such a way that the inner product of the latent features of a user with the latent feature of an item is equal to that user's rating on that item. In reference [16], after extracting the latent features obtained by matrix factorization, the support vector machine model [17] was used to classify items of each user into two categories of interesting and non-interesting items. To strengthen the suggestions of matrix factorization based recommendation system, a deep-convolutional neural network can be used [18]. The role of the convolutional neural network is to help to extract deep latent features of items based on the comments of experts or customers on items.

As mentioned, one of the most successful types of recommender systems is the M-distance based recommendation system (MBR) which uses the average ratings given to an item as the attribute of that item. This attribute is used to determine similar items. In this paper, to decrease the error of MBR, eight recommendation systems are proposed based on the MBR method, which combines the following ideas: (a) Using the variance of item ratings in addition to the average of item ratings, as two attributes of item, for determining similar items in an item-based nearest neighbor method; (b) Using the variance of user ratings in addition to the average of user ratings, as two attributes of user, for determining similar users in a user-based nearest neighbor method; (c) Using a weighted average method for combining the ratings of similar items or similar users; (d) Using ensemble learning. Experimental results on real datasets show that our proposed EVMBR and EWVMBR which use ensemble learning have the least error. For different datasets, the mean absolute error of the suggested EWVMBR method is at-least 20 percent lower than that of MBR, Slope-One, P-kNN, and C-kNN. The runtime of our proposed methods is competitive with the MBR runtime and much lower than Slope-One, P-kNN, and C-kNN.

The main contribution of this paper is using variance of rating as a new feature of items and users. Indeed, we defined a new distance metric called VM-distance by using the variance statistics to compare two items or two users and to find the nearest neighbors of an item or a user. We also proposed a weighted average method for combining the ratings of similar items or similar users;

In continue, in section 2, the M-distance based recommendation system (MBR) is explained in detail. In section 3, our proposed methods are presented. In section 4, the experimental results are presented, and in section 5 the conclusion is drawn.

2. MBR

In MBR, which is a nearest neighbor based method, the average of ratings given to each item is used as a feature of that item to determine similar items. Formally, the average rating of the i-th item is given by

$$\mu_i = \frac{\sum_{u=1}^{n} s_{ui}}{|\{s_{ui} \mid s_{ui} \neq 0, 1 \leq u \leq n\}|} \quad (1)$$

where n is the number of users in a user-item matrix, and s_{ui} is the rating given to the item i by the user u. As you can see, in the calculation of the average using Eq. (1), the missing ratings marked with zero are not considered. The M-distance of item i and item k is defined as follows:

$$md_{i,k} = abs(\mu_i - \mu_k). \quad (2)$$

The MBR algorithm is summarized as algorithm 1. In this algorithm, for predicting the rating of user u on item i, first, some of the nearest neighbors of the item i which were rated by the user u are determined. Then, if there exists at least one neighbor, the predicted rating of the user u on item i (p_{ui}) is set to the average of ratings of the user u on those nearest neighbors. Otherwise, the value of p_{ui} set to the average the ratings of the different users on the item i. In other words,

$$p_{ui} = \begin{cases} mean(\{s_{uk} | md_{i,k} \leq \delta\}) & |\{k | md_{i,k} \leq \delta\}| > 0, \\ \mu_i & otherwise. \end{cases} \quad (3)$$

where $\{k | md_{i,k} \leq \delta\}$ is the set of those items of which M-distances to i-th item is less than the threshold δ.

Algorithm 1. MBR algorithm [4].

Input:

δ: The neighborhood threshold
s: User-item matrix
m: The number of items
u: The active user
i: The item of the active user which its rating must be predicted

Output:

p_{ui}: The predicted rating of user u on item i

/* Step 1. Find nearest neighbor of item i */
$nb = 0$ // Number of nearest neighbors of item i
$nbsum = 0$ // The sum of ratings of nearest neighbors
$for\ k = 1\ to\ m$
  /* an item cannot be the neighbor of itself*/
    $if\ (k == i)\ then$
      Continue;
    $endif$

/* Items with zero ratings are not considered as neighbors*/

$if\ (s_{uk} == 0)\ then$

continue;

$endif$

$md_{ik} = abs(\mu_i - \mu_k)$

$if\ (md_{ik} \leq \delta)\ then$

$nb++$

$nbsum += s_{uk}$

$endif$

$endfor$

/*Step 2. Predicting the rating of active user on item i */

$if\ (nb \geq 1)\ then$

$p_{ui} = \frac{nbsum}{nb}$

else

$p_{ui} = \mu_i$

$endif$

end

3. OUR PROPOSED METHODS

In this paper, eight recommendation systems are proposed based on the MBR method. The idea of these eight recommendation systems are as follows:

- Using the variance of item ratings in addition to the average of item ratings, as two item attributes, for determining similar items in an item-based nearest neighbor method.
- Using the variance of user ratings in addition to the average of user ratings, as two user attributes, for determining similar users in a user-based nearest neighbor method.
- Using a weighted average for combining the ratings of similar items or similar users.
- Using ensemble learning.

3.1 Item-based methods

The MBR method is an item-based nearest neighbor method. In this sub-section, based on this item-based method, several item-based methods are suggested.

3.1.1 Using the variance statistics

The variance of the ratings of item i is defined as follows:

$$v_i = \frac{\sum_{u=1}^{n}(s_{ui}-\mu_i)^2}{|\{s_{ui} \mid s_{ui}\neq 0, 1\leq u\leq n\}|-1} \quad (4)$$

where s_{ui} is the rating given by the user u on the item i. As it can be seen, the missing ratings (marked with zero) are not considered for computing the variance. μ_i is the average ratings of users on the item i.

Definition. The VM-distance of item i and item k is defined as follows:

$$vmd_{i,k} = \text{abs}(\mu_i - \mu_k) + \alpha \times abs(v_i - v_k), \quad (5)$$

where the parameter $\alpha \geq 0$ determines the importance of the variance with respect to the average.

3.1.2 The weighted average of neighbors' ratings

In the MBR method, the un-weighted average of the ratings of a number of similar rated items to an unrated item of the active user is used to determine the rating of that unrated item. Using the weighted average instead of an un-weighted average may increase the accuracy of the MBR algorithm. Therefore, to predict the active user rating on the item i, the following equation is proposed:

$$p_{ui} = \begin{cases} \frac{\sum_{\{k|vmd_{i,k}\leq\delta\}} w_k s_{uk}}{\sum_{\{k|vmd_{i,k}\leq\delta\}} w_k} & \left|\{k|vmd_{i,k}\leq\delta\}\right| > 0, \\ \mu_i & otherwise, \end{cases} \quad (6)$$

where $w_k > 0$ is the Gaussian weight of item i, namely

$$w_k = \exp\left(\frac{-vmd_{i,k}}{\sigma^2}\right), \quad (7)$$

where $\sigma > 0$ is Gaussian function width. The greater the $vmd_{i,k}$ (VM-distance between item i and item k) is, the less the effect or the weight of item k (w_k) for predicting the rating of user u on item i is. The parameter σ determines how the weight changes when the VM-distance changes. For example, if σ is small, the weight tends to zero rapidly as the VM-distance increases, while if σ is big, the weight tends to zero slowly as the VM-distance increases.

3.2 User-based methods

The MBR method is an item-based nearest neighbor method. In continue, based on this item-based method, several user-based methods are suggested.

3.2.1 Using the variance statistics

The average and the variance of the ratings of user u is determined as follows:

$$\tilde{\mu}_u = \frac{\sum_{i=1}^{m} s_{ui}}{|\{s_{ui} \mid s_{ui}\neq 0, 1\leq i\leq m\}|} \quad (8)$$

$$\tilde{v}_u = \frac{\sum_{i=1}^{m}(s_{ui}-\tilde{\mu}_u)^2}{|\{s_{ui} \mid s_{ui}\neq 0, 1\leq i\leq m\}|-1} \quad (9)$$

where m is the number of items in the user-item table.

The VM-distance of user u and user t is given by

$$\widetilde{vmd}_{u,t} = \text{abs}(\tilde{\mu}_u - \tilde{\mu}_t) + \alpha \times abs(\tilde{v}_u - \tilde{v}_t), \quad (10)$$

where the parameter $\alpha \geq 0$ determines the importance of the variance with respect to the average.

3.2.2 The weighted average of neighbours' ratings

Using the weighted average instead of an un-weighted average may increase the accuracy of the nearest neighbor algorithm. Therefore, to predict the active user rating on the item i, the following equation is proposed:

$$p_{ui} = \begin{cases} \frac{\sum_{\{t|\widetilde{vmd}_{u,t}\leq\delta\}} \tilde{w}_t s_{ti}}{\sum_{\{t|\widetilde{vmd}_{u,t}\leq\delta\}} \tilde{w}_t} & \left|\{t|\widetilde{vmd}_{u,t}\leq\delta\}\right| > 0, \\ \tilde{\mu}_u & otherwise, \end{cases} \quad (11)$$

where $\tilde{w}_t > 0$ is the Gaussian weight of the user t, namely

$$\tilde{w}_t = \exp\left(\frac{-\widetilde{vmd}_{u,t}}{\sigma^2}\right). \quad (12)$$

3.3 Ensemble learning based recommendation system

One way to improve the accuracy of predictive machines is to use ensemble learning. In this method, several predictive machines are used to predict. Then, the results of the predictive machines are combined. In this paper, an item-based method and a user-based approach are used as two members of our proposed ensemble learning. In other words, after estimating the ratings with each of these two learning methods, the weighted average of these two estimated ratings is reported as the final value of the estimated rating. The accuracy of each learning method is the weight of that method in ensemble learning which is determined using a validation dataset.

Finally, the eight proposed recommendation systems of this paper are summarized as follows:

- VMBR-I: An item-based approach which uses our proposed VM-distance criterion to calculate the difference between two items and to determine the similar items. It then uses un-weighted average of the ratings of a number of nearest items of an unrated item of an active user as the estimated rating of that unrated item.
- VMBR-U: A user-based approach which uses our proposed VM-distance criterion to calculate the difference between two users and to determine the similar users. It then uses un-weighted average of the ratings of a number of nearest users on an active item as the rating of an active user on the active item.
- WVMBR-I or Weighted VMBR-I: An item-based approach which uses our proposed VM-distance criterion to calculate the difference between two items and to determine the similar items. It then uses the weighted average of the ratings of a number of nearest items of an unrated item of an active user as the estimated rating of that unrated item.
- WVMBR-U or Weighted VMBR-U: A user-based approach which uses our proposed VM-distance criterion to calculate the difference between two users and to determine the similar users. It then uses the weighted average of the ratings of a number of nearest users as the rating of an active user on an active item.
- WMBR-I or Weighted MBR-I: An item-based approach which uses the M-distance criterion to calculate the difference between two items and to determine the similar items. It uses the weighted average of the ratings of a number of nearest items as the rating of an unrated item of the active user.
- WMBR-U or Weighted MBR-U: A user-based approach which uses the M-distance criterion to calculate the difference between two users and to determine the similar users. It uses the weighted average of the ratings of a number of nearest users as the rating of an active user on an active item.
- EVMBR (VMBR with Ensemble Learning): A recommendation system which uses ensemble learning, and VMBR-I and VMBR-U as two members of the ensemble learning method.
- EWVMBR (WMBR with Ensemble Learning): A recommendation system which uses ensemble learning, and WVMBR-I and WVMBR-U as two members of the ensemble learning method.

4. EXPERIMENTAL RESULTS

In this section, our proposed methods are evaluated using a number of experiments and are compared with the MBR methods, cosine-based kNN [1], Pearson-based kNN, and Slope-One [5]. The effectiveness of our proposed methods is evaluated by using three datasets, i.e. Movielens-100k, DouBan, and EachMovie. MovieLens-100K consists of 100,000 ratings from 943 users on 1,682 movies. DouBan consists of 912,479 ratings from 2,925 users on 39,695 movies. EachMovie consists of 913,809 ratings from 72,916 users on 1,628 movies. All experiments were performed on a computer of 3.1GHz CPU and 12GB memory.

Leave-one-out method is used to evaluate the proposed algorithms. The most common error estimation criteria, i.e. the mean absolute error (MAE) and root mean square error (RMSE), is used to report the error rate of every method.

Table 1 compares the MAE of different recommendation methods for the best value of threshold δ. Bold font indicates the best algorithms. As it can be seen, our proposed EWVMBR has the least MAE for each datasets. The MAE of each of our proposed methods is less than that of MBR, P-kNN, C-kNN, and Slope-One methods. The MAE of the EWVMBR method is 20 to 30 percent lower than that of the MBR for each datasets.

Table 1. The best MAE of different recommendation systems

	MovieLens100K	DouBan	EachMovie
P-kNN	0.8363	0.7089	0.2277
C-kNN	0.7487	0.6366	0.1980
Slope-One	0.7421	0.5902	0.2900
MBR	0.7389	0.5869	0.1933
VMBR-I	0.6702	0.5296	0.1410
VMBR-U	0.7062	0.5794	0.1503
WMBR-I	0.6175	0.4673	0.1294
WMBR-U	0.6572	0.5119	0.1437
WVMBR-I	0.6165	0.4637	0.1294
WVMBR-U	0.6565	0.5116	0.1635
EVMBR	0.6519	0.5310	0.1415
EWVMBR	**0.5970**	**0.4636**	**0.1292**

Table 2. The best RMSE of different recommendation systems

	MovieLens 100K	DouBan	EachMovie
P-kNN	1.044	0.8859	0.2847
C-kNN	0.9652	0.8150	0.2567
Slope-One	0.9432	0.7458	0.3318
MBR	0.9412	0.7479	0.2508
VMBR-I	0.9271	0.7059	0.1891
VMBR-U	0.9694	0.7772	0.1894
WMBR-I	1.0289	0.7788	0.2117
WMBR-U	1.0627	0.8372	0.1920
WVMBR-I	1.2095	0.7806	0.2117
WVMBR-U	1.0627	0.8370	0.1919
EVMBR	**0.8533**	0.6834	0.1800
EWVMBR	0.8605	**0.6680**	**0.1743**

Tables 2 illustrates the RMSE of the different recommendation methods for the best value of threshold δ. Bold font indicates the best algorithms. As can be seen, our proposed EWVMBR or EVMBR has the least RMSE for all three datasets. Also, the RMSE of our proposed VMBR-I is also less than that of the traditional recommendation methods, i.e. MBR, P-kNN, C-kNN, and Slope-One. The RMSE of our

proposed EWVMBR and EVMBR are 10 to 20 percent lower than the MBR method for all three datasets.

Table 3 shows the runtime time of different methods for the best threshold value δ. Our proposed VMBR-I or VMBR-U has the least runtime for all three datasets. The VMBR-U which is a user-based method has the least runtime for the MovieLens100K and DouBan datasets. The VMBR-I which is an item-based method has the least runtime for the EachMovie dataset. The reason is that the number of users is less than the number of items in MovieLens100K and DouBan datasets, while in the EachMovie dataset, the number of users is more than the number of items. Therefore, in the EachMovie dataset with 72916 users and 1628 items, the runtime of determining the nearest items is much less than that of the nearest users' determination. Notice that VMBR-I searches for the nearest items while VMBR-U method search for the nearest users.

The only difference between the VMBR-I method and the MBR method is the use of the variance of item ratings in addition to the mean of item ratings in the VMBR-I method. Therefore, it seems that the runtime of the VMBR-I method must be more than that of the MBR, while Table 3 does not confirm this. The reason is that the reported runtime is the runtime for the optimum threshold δ. The threshold δ directly relates with the numbers of nearest neighbors used to predict a rating. The optimum thresholds of different recommendation methods are not the same. The optimum threshold value δ of VMBR-I and MBR are 0.01 and 0.5, respectively. Therefore, the number of neighboring items used in the VMBR-I method is less than that of the MBR which leads the runtime of the VMBR-I method to be also less than that of MBR.

Table 3. Runtime (seconds)

	MovieLens 100K	DouBan	EachMovie
P-kNN	410.2365	120765.6565	21087.7639
C-kNN	399.6544	118654.7629	20546.6532
Slope-One	397.2097	117233.7626	20351.6334
MBR	2.3885	6543.0187	1072.4803
VMBR-I	0.5260	4040.3133	**106.7081**
VMBR-U	**0.2961**	**66.5703**	584.4321
WMBR-I	3.1222	8569.7917	1271.8774
WMBR-U	3.0339	835.3166	7145.0752
WVMBR-I	2.9670	4755.3064	1232.4631
WVMBR-U	3.2888	814.7281	7113.4829
EVMBR	0.6877	4.93.5993	709.9855
EWVMBR	6.5287	9444.0830	8374.5360

Figure 1 shows the sensitivity of the MAE of the WVMBR-I to the threshold δ for the Movielens-100k dataset. In this experiment, the value of the parameter σ is considered to be 10^{-4}. As it can be seen, the error decreases by increasing the threshold δ. According to the Figure 2, the runtime is increased as the threshold δ increases. The reason is that as the threshold δ increases the number of neighbors which are used to predict the ratings increases. The threshold parameter δ in an un-weighted method, such as MBR or VMBR-I, or VMBR-U, is the only parameter for controlling the number of contributing neighbors used for predicting ratings. But, in the WVMBR-I method, which is a weighted method, the parameter of Gaussian weight, i.e. σ, can also control the effects of neighbors, and can eliminate the effect of distant neighbors.

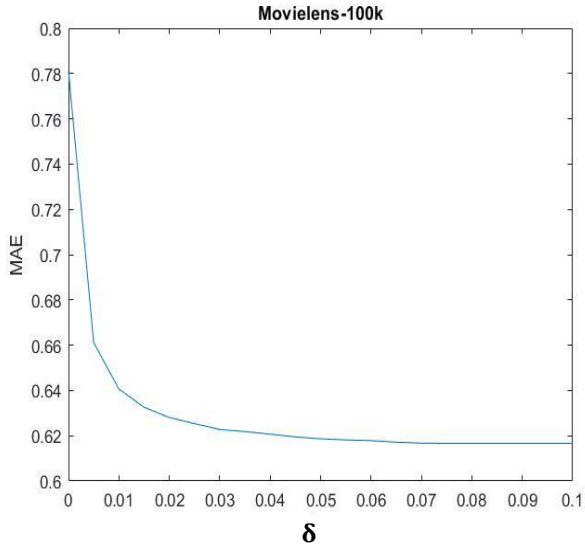

Figure 1. The MAE sensitivity of WVMBR-I method to the threshold value δ for Movielens-100k dataset

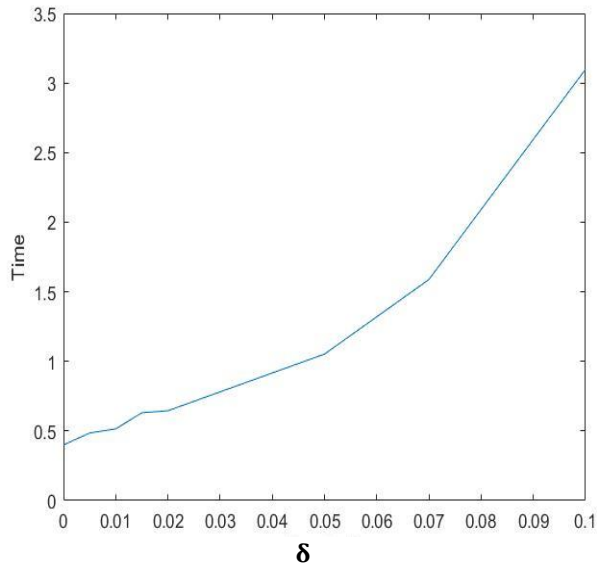

Figure 2. The runtime sensitivity of WVMBR -I method to the threshold value δ for Movielens-100k dataset

Figure 3 shows the MAE sensitivity of the WVMBR-I method to the value of the parameter σ for the Movielens-100k dataset. In this experiment, the threshold δ is considered to be 0.1. By increasing the parameter σ from 0 to 0.0001, the MAE decreases, then the MAE increases as the parameter σ increases. The parameter σ determines how the weight of a neighbor changes when its VM-distance changes. For example, if σ is small, the weight tends to zero rapidly as the VM-distance increases, while if σ is big, the weight tends to zero slowly as the VM-distance increases. Therefore, if σ is small, for example 0.0001, only the most similar items are used for rating prediction. If σ is very small, for example 0.00001, a zero weight may be assigned to even the nearest neighbor which is a problem for predicting a rating. If σ is big, for example 1, even distant neighbor is used to predict rating. Since distant neighbor of an item are not similar to that item, using distant neighbor can increase prediction error.

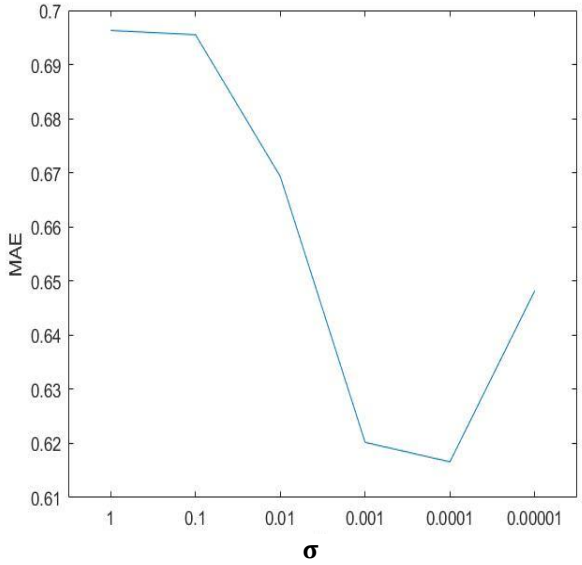

Figure 3. The MAE sensitivity of WVMBR-I method to the parameter value **σ** for Movielens-100k dataset

5. CONCLUSION

In this paper, eight recommendation systems were proposed based on the MBR method, which combines the following ideas:

(1) Using the variance of item ratings along with the average of item ratings, as two attributes of the item, for determining similar items in an item-based nearest neighbor method;

(2) Using the variance of user ratings along with the average of user ratings, as two attributes of the user, for determining similar users in a user-based nearest neighbor method;

(3) Using a weighted average method for combining ratings of similar items or similar users;

(4) Using ensemble learning.

Experimental results showed that our proposed EVMBR and EWVMBR methods which use ensemble learning have the least error. For different datasets, the mean absolute error of the suggested EWVMBR method is at-least 20 percent lower than that of MBR, Slope-One, P-kNN, and C-kNN. The runtime of the proposed methods is competitive with the MBR runtime and much lower than Slope-One, P-kNN, and C-kNN. Meanwhile, our proposed user-based methods have less runtime than that of our proposed item-based methods for the dataset which the number of its users are less than the number of its items.

In the future, in order to improve the accuracy of MBR-based methods, along with the mean and variance statistics, we can use other statistics such as the average of each quartet of item (or user) ratings as the other attributes of item (or user).

REFERENCES

[1] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web, Hong Kong, pp. 285-295.

[2] Ahn, H.J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences, 178(1): 37-51. https://doi.org/10.1016/j.ins.2007.07.024

[3] Park, Y., Park, S., Jung, W., Lee, S.G. (2015). Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph. Expert Systems with Applications, 42(8): 4022-4028. https://doi.org/10.1016/j.eswa.2015.01.001

[4] Zheng, M., Min, F., Zhang, H.R., Chen, W.B. (2016). Fast recommendations with the m-distance. IEEE Access, 4(1): 1464-1468. https://doi.org/10.1109/ACCESS.2016.2549182

[5] Lemire, D., Maclachlan, A. (2005). Slope one predictors for online rating-based collaborative filtering. Proceedings of the 2005 SIAM International Conference on Data Mining, Newport Beach, pp. 471-475.

[6] Li, J., Sun, L., Wang, J. (2011). A slope one collaborative filtering recommendation algorithm using uncertain neighbors optimizing. International Conference on Web-Age Information Management, Berlin, pp. 160-166.

[7] Wang, Q.X., Luo, X., Li, Y., Shi, X.Y., Gu, L., Shang, M.S. (2018). Incremental slope-one recommenders. Neurocomputing, 272(1): 606-618. https://doi.org/10.1016/j.neucom.2017.07.033

[8] Bu, J., Shen, X., Xu, B., Chen, C., He, X., Cai, D. (2016). Improving collaborative recommendation via user-item subgroups. IEEE Transactions on Knowledge and Data Engineering, 28(9): 2363-2375. https://doi.org/10.1109/TKDE.2016.2566622

[9] Salter, J., Antonopoulos, N. (2006). CinemaScreen recommender agent: Combining collaborative and content-based filtering. IEEE Intelligent Systems, 21(1): 35-41. https://doi.org/10.1109/MIS.2006.4

[10] Van Meteren, R., Van Someren, M. (2000). Using content-based filtering for recommendation. Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, Barcelona, pp. 47-56.

[11] Yao, Y. (2010). Three-way decisions with probabilistic rough sets. Information Sciences, 180(3): 341-353. https://doi.org/10.1016/j.ins.2009.09.021

[12] Yao, Y. (2015). Rough sets and three-way decisions. International Conference on Rough Sets and Knowledge Technology, Tianjin, pp. 62-73.

[13] Qi, J., Qian, T., Wei, L. (2016). The connections between three-way and classical concept lattices. Knowledge-Based Systems, 91(1): 143-151. https://doi.org/10.1016/j.knosys.2015.08.006

[14] Condli, M.K., Lewis, D.D., Madigan, D., Posse, C. (1999). Bayesian Mixed-E ects Models for Recommender Systems. ACM SIGIR, vol. 99.

[15] Yuan, Y., Luo, X., Shang, M.S. (2018). Effects of preprocessing and training biases in latent factor models for recommender systems. Neurocomputing, 275(1): 2019-2030. https://doi.org/10.1016/j.neucom.2017.10.040

[16] Ren, L., Wang, W. (2018). An SVM-based collaborative filtering approach for Top-N web services recommendation. Future Generation Computer Systems, 78(1): 531-543. https://doi.org/10.1016/j.future.2017.07.027

[17] Vapnik, V. (1995). The Nature of Statisticsal Learning. ed: Springer, NY.
[18] Wu, H., Zhang, Z., Yue, K., Zhang, B., He, J., Sun, L. (2018). Dual-regularized matrix factorization with deep neural networks for recommender systems. Knowledge-Based Systems, 145(1): 46-58. https://doi.org/10.1016/j.knosys.2018.01.003

NOMENCLATURE

μ_i	average rating of the i-th item
n	number of users in a user-item matrix
u	active user
i	an item of the active user of which rating must be predicted
s_{ui}	rating given to the item i by the user u.
$md_{i,k}$	M-distance of item i and item k
δ	a threshold to determine nearest neighbours of an item
p_{ui}	predicted rating of the user u on item i
v_i	variance of rating of the i-th item
$\tilde{\mu}_u$	average rating of the user u
$\tilde{v}_u$	variance of rating of the user u
$vmd_{i,k}$	VM-distance of item i and item k
w_k	Gaussian weight of item i
$\widetilde{vmd}_{u,t}$	VM-distance of user u and user t
$\tilde{w}_t$	Gaussian weight of user t