



DBT Recommender: Improved Trustworthiness of Ratings through De-Biasing Tendency of Users

Mukkamula Venu Gopalachari^{1*}

¹*Department of Computer Science and Engineering,
Chaitanya Bharathi Institute of Technology, Hyderabad, India*

* Corresponding author's Email: venugopal.m07@gmail.com

Abstract: Recent advancements in business strategies marked the significance of e-commerce in marketing any service of the organization. Moreover, users are quite dependent on the average ratings of the products showcased in the marketing interface in turn these average ratings made remarkable impact on sales phenomena of the product. The average rating of a product is the aggregation of individual users ratings biased with the tendency of the user towards publishing the opinion. The optimistic user tends to give a slight high rating than a neutral judgement and vice versa with a pessimistic user. However, these biased ratings produce an aggregate value that is degraded with its trustworthiness. This paper proposed a novel approach named DBT (De-biased Tendency) Recommender to analyze the bias in product rating which recalculates the average ratings of the products by making user tendencies as part of the process. The solution implemented on a big data environment on demand of high computation complexity involved in the process. Experimental results had shown a significant improvement in the trustworthiness of the product ratings with the proposed approach.

Keywords: De-biasing, E-commerce, Opinion mining, Recommender system.

1. Introduction

E-commerce is one of the sales sectors that grabbed huge part of the market scope since past decade. The enormous popularity of the e-commerce sales is mostly due to the crowd sourcing phenomena across the customer community through various mediums. One of the main crowd sourcing components that increases the popularity of a product in an e-commerce portal is “feedback” and also known as “rating” which generally ranges from numeric 1 to 5 in the ascending satisfaction metric representation. This rating metric exactly represents the opinion of the particular user towards a product or service purely subject to the interest of the customer. But these opinions as an aggregated value advertises the product to the new customers that impacts as first impression towards the transaction. The average ratings values reveal information about peers after a customer visits with own opinion which also improves the transparency in encouraging the

political engagements on platforms of open democracy [1]. However, the ratings provided by the previous customers might be biased by factors such as their natural optimistic level of opinion, current emotional status, interestingness about feedback process etc. Social influence bias can yield ratings that are closer to the average, less diverse and in turn less representative of participants' true evaluations for items, which can in turn produce bias in similarity measures between items and users.

The online recommendations are the recent trendsetter that affects not only the willingness to make transaction but also the consumer's preference ratings [2]. Recent studies found the evidence about recommendations that an online system serves as an anchor when consumers form their preference for products, even at the time of consumption on TV shows, songs and jokes datasets [3]. These kind of biased ratings of the users as crowd sourcing would lead to the potential issues such as a) could lead to

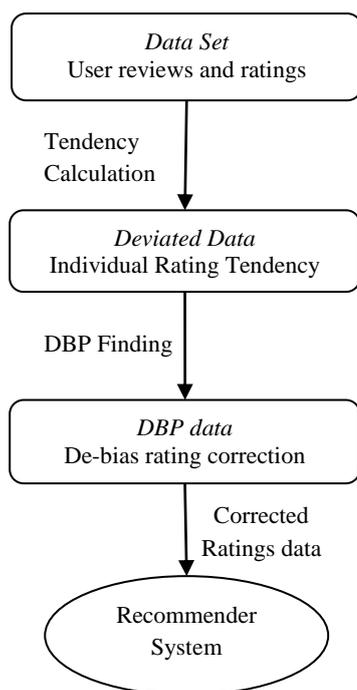


Figure.1 Architecture of the DBT recommender

the cold item problem with the distorted view from the customers, b) reducing the recommender's ability to serve the quality recommendations to the future customers, c) could leave wrong impression to the customers about the recommender performance with distorted suggestions [3].

This tendency leads to the lack of trustworthiness in the average ratings provided as basic information. Thus the handling of the biased ratings is necessary that defines a potential research problem to be resolved. The possible solution is to approximate average ratings by de-biasing the ratings given by the individual users. Some of the researchers tried to resolve the problem by analyzing the sentiment level of the review text using various classification methods [4]. However, the techniques involved high computational complexity which lags the performance of the system.

In this paper, a recommender system (DBT Recommender) is proposed to improve the quality of trustworthiness of the recommendations using a novel personalized approach to de bias the individual ratings works as shown in Fig. 1. This approach defines a model that gives the De-Biased Proportionate (DBP) value of a particular user based on the history of personalized ratings and average rating of that product. These DBP values would modify the future ratings of the user. The proposed recommender is implemented as a big data application as it involved with huge amount of

computations with personalization. Experimental results have shown better quality than existing de-biasing methods on the benchmark test data sets.

The rest of the paper is organized as follows. The related work is presented in the next section and then the architecture of the proposed DBT Recommender is explained. There after results and analysis shows the experimentation part followed by conclusion.

2. Related work

Recommender Systems (RS) are the software agents target to predict the preferences of the user and serve accordingly. RS is uses by many areas in web such as movies [5], music [6], news [7], tourism [8], social networks [9] and scientific papers [10]. The main approaches of the recommenders includes content based and collaborative filtering where the first one predicts the interests of the user based on the sole user's data whereas the later one predicts the interests of the user based on the similar user's data[11]. The main focus of the proposed work is on Collaborative Filtering (CF), which plays an important role in RS, since it has been one of the most successful methods of recommendation. CF techniques mainly classified as two different categories, known as Memory-Based CF and Model-Based CF. Memory-Based CF provides recommendations based on similarities among users or items in turn predictions are made for the target user using entire user-item database. On the other hand, model-based CF approaches uses the user-item database to generate a learning or statistical model [12].

The popular category of algorithms in CF is memory based algorithms which takes high computation complexity and widely used in the industry. This kind of recommenders uses user-item matrix in processing the prediction of ratings for the future access. When compared with other kinds of CF techniques memory based techniques are easy to implement and understand because it works without explicit knowledge of the model beneath and of course preferable by the e-commerce systems such as Google, Netflix and Amazon [13].

However, the traditional CF systems suffer from issues such as cold-start and data sparsity problems. Cold-start problem occurs when a user has less history to predict preferences and data sparsity occurs with the nature of ability of a user to rate the items. Cheng et al. developed a fuzzy RS to overcome the weakness of CF, which uses opinions of the users in the fuzzy linguistic model to clarify the preferences of the user based on similar users'

opinions [14]. Boratoo et al. proposed GRS (Group Recommender System) which makes predictions on missing ratings to overcome sparsity before clustering users to find neighbors. But, this approach leads to high time complexity due to the similarity computations on large data [15]. After that, Ghazarian et al. in [16] used SVM regression in order to train a model to compute similarities on the item's features. The results achieved from item similarity calculation were then used to make predictions on the missing values of the matrix. Rendle combined the advantages of SVM with factorization models and came up with an algorithm called Factorization Machines (FM) [17]. The algorithm can solve problems with huge sparsity where SVMs usually fall short. For data with strong relational patterns, the feature vector of a data instance can get excessively large. This will lead to learning and prediction becoming slow or even not practical. So [18] scaled FM to relational data by making use of recurrent patterns in the feature vectors.

Xu et al. in [19] recommended nonnegative matrix completion (NMC) in order to solve the sparsity problem, where the aim is to introduce an intermediate, complete matrix to estimate a target rating matrix, and use an NMF process on this matrix instead of the incomplete rating matrix to avoid addressing its missing entries. Though these models are good to deal with the incomplete one in a CF problem, they have the drawback of high computational and storage costs, which are linear with the size of the target matrix. Zhou et al. in [20] recommend an incremental approach based on SVD that constantly computes the singular value decomposition of the original matrix unchanged each time to solve the sparsity problem and users' interests that are dynamic. Furthermore, other data mining techniques including clustering [21], classification [22], and association rules mining [23] have been used in recommender systems as solutions to the data sparsity.

The stated biasing problem was coined as social influence bias, which impacts the opinion of the user by the other user's influence, in the literature. Also the rating bias according to timeline is experimented in [24], proving that the rating at the beginning and later parts of the product's lifetime by the same user is different. Adomavicius et al. (2013) looked at a similar effect in an even more controlled setting, in which the consumer preference ratings for items were elicited at the time of item consumption. Nagle et.al. proposed that the rating of the user represents

the opinion of the user as well the current state of the reviews of the product [25].

Muchnik et al. designed a randomized experiment in which comments in an online forum were randomly up-treated or down-treated and concluded a bias where a positive rating trend improved the positive rating likelihood by 32%. The researchers concluded the priori setting problem that users see the aggregate rating before giving their rating [26]. But these experiments did not able to raise a solution for the future biases in recommender system. Zhu et al. conducted an experiment in which users evaluate an image on a subjective question with binary scale (e.g. Is this image cute?), which was followed by a presentation of the crowd consensus opinion [27]. Along these lines, Sipos et al. argue that context along with an aggregate rating plays a large role in the users' ratings. That is, users may attempt to correct the average, by voting in a more polarizing manner (more positively or negatively). Prior research also found that recommendations not only significantly affect consumers' preference ratings but also their economic behavior. Krishnan et al. suggest introducing the 3-step rating system applied by their case-study, the California Report Card, and then use machine learning to estimate the social influence bias and to correct it before the review is posted onto the platform [28]. The proposed approach is different from the existing work due to the personalization applied in de-biasing the individual ratings according to the existing rating information.

3. DBT recommender

The architecture of the proposed de-biasing approach is shown in Fig. 1. This approach has two major parts in its process: i) finding tendency of the customer ii) Calculation of DBP value to correct the ratings of the user. First step is a personalization process that collects the history of the individual users and defines the tendency of the user as how much positive or negative by nature. The later step is to define DBP value that actually used to correct the ratings of the individual customer. The outcome of the two steps contributes the input for the recommender system by which the recommendations would be provided to the user. For example if a user had a positive tendency in rating of the value numeric one means the actual rating that can be allotted must be the rating given by the user subtracted by numeric one.

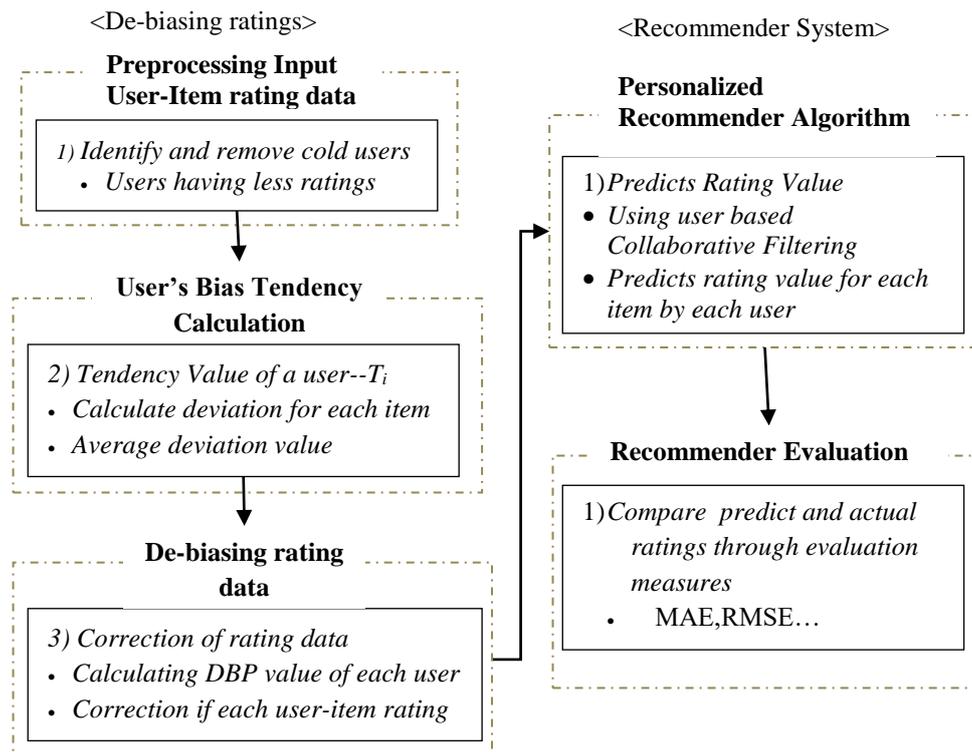


Figure.2 Process flow of DBT recommender system

3.1 Tendency calculation

The personalized ratings of the user are processed against the aggregate ratings in order to find the tendency of the user. In this process the cold users are omitted because the tendency of the cold user could not be found with insufficient rating data. The tendency of a user U_i is calculated using the rating data $R_i = \{r_1, r_2 \dots r_j\}$ of 'j' items and aggregated ratings data of the items $AR_i = \{ar_1, ar_2 \dots ar_j\}$ as in the following steps.

Step 1: Count the number of items engaged and if count is less than a predefined threshold (τ) then treat user as cold user and read next user data.

if (count(R_i) $\geq \tau$) then go to step 2

Step 2: Calculate the deviation of the user rating from the aggregate rating for all the items engaged.

$$D_p = \{d_1, d_2 \dots d_m\}$$

$$D_n = \{d_1, d_2 \dots d_n\}$$

where $d_k = r_k - ar_k$,
 if $ar_k \geq r_k$ then $d_k \in D_n$
 Otherwise $d_k \in D_p$

Step 3: Calculate the average deviation of the user for all the items engaged based on the positive or negative tendency of the customer deviations.

$$\text{if } (|m - n| > \beta) \text{ then}$$

$$\text{if } (m > n) T_i = \frac{\sum_{l=1}^m d_l}{m}$$

$$\text{else if } (m < n) T_i = \frac{\sum_{l=1}^n d_l}{n}$$

else
 $T_i = 0$;

After these steps T_i value, which represents the tendency of the customer in rating items, is given to the next part of the proposed approach. If T_i is positive then the customer used to give ratings positively than the actual quality of the product and vice versa. The bigger value of the T_i represents the biasing grade of the user in rating.

3.2 DBP finding

DBP value of the customer is found based on the tendency value T_i found in the previous step. However the proposed approach uses a predefined de biasing factor (α) that calculates the DBP value from the tendency of the user. The derived individual de-biasing factor would be applied to correct all the previous and future ratings of the user.

The set $DBP = \{DBP_1, DBP_2 \dots DBP_N\}$, where DBP_i is the DBP value of the individual user is the product of the de-biasing factor (α) and the tendency of the user (T_i).

```

for each  $U_i$  in  $U = \{U_1, U_2 \dots U_N\}$ 
if ( $T_i > 0$ ) then
     $DBP_i = \alpha \times T_i$ ;
else
     $DBP_i = 0$ ;
return DBP.

```

Thus the DBP values of the users were calculated and forwarded to the next step in the process.

The steps to correct the ratings in the rate matrix returns the modified ratings of all users $MR = \{MR_1, MR_2 \dots MR_N\}$ according their respective tendencies $T = \{T_1, T_2 \dots T_N\}$ where each user modified rating set $MR_i = \{mr_1, mr_2 \dots mr_j\}$ consists ratings of 'j' items. Pseudo code to correct the individual ratings of users U according to individual DBP is as follows:

```

Input:  $MR_i = \{mr_1, mr_2 \dots mr_j\}$ 
for each  $U_i$  in  $U = \{U_1, U_2 \dots U_N\}$ 
if ( $T_i = 0$ ) then
    for each  $r_c$  rating of  $R_i = \{r_1, r_2 \dots r_M\}$ 
         $mr_c = r_c$ ;
else
    for each  $r_c$  rating of  $R_i = \{r_1, r_2 \dots r_M\}$ 
        if  $r_c$  exists then
             $mr_c = r_c - DBP_i$ ;
return MR.

```

Thus the modified ratings are stored as input rating matrix for serving the recommender system.

3.3 DBT recommender

DBT Recommender is the recommender system that uses the Modified Ratings to define the de-biased suggestions to the user. Now the de-biased modified ratings of the each user on the concerned items are ready for the recommender application. From the available recommenders, the suitable one for the Modified Rating data set must be identified based on the performance requirement of the recommender application. The proposed methodology used collaborative filtering technique to develop recommender system. The proposed algorithm implemented collaborative filtering technique on Modified Rating matrix to find the recommendations.

Collaborative Filtering (CF) is one of the most frequently used RS techniques, in which, items are recommended to the target user through an analysis of most similar users' (neighbor users) ratings on those items[11]. This method can be classified into user-based CF and item-based CF. In the user-based approach, it is assumed that if some users have similar interests, they will have similar interests in the future as well. Based on this assumption, items are recommended to the target user. In the item-based algorithms, the similarity between items is looked into in order to make predictions. The general perception is that a user will most probably purchase items similar to the ones he bought in the past.

User Based CF which falls under the category of Memory based CF operates on an $n \times m$ user– item matrix. The matrix records the preferences of n users on m items. When a new user, known as the target user, enters the system and demands recommendations, users most similar to the target user, known as neighbor users, are determined in the system. Through looking into their earlier ratings on that specific item, a prediction will be made for the target user with regard to that particular item. In other words, recommended items will be those previously chosen by users having a similar taste as the target user.

Items recommended to the target user will be chosen on the basis of the preferences of neighbor users. One of the common methods used in CF to find neighbor users, is through resorting to similarity measures [11]. A variety of similarity measures can be found in the literature like Pearson Correlation Coefficient (PCC), Cosine, Spearman, etc. [30]. However, empirical analyses indicate that PCC outperforms other similarity measures in user-based CF[12]. The PCC between users a and b can be measured through Eq. (1).

$$sim(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}} \quad (1)$$

PCC range is [-1, 1], -1 means complete dissimilarity and 1.0 denotes complete similarity. Negative values may reduce the recommendation accuracy; therefore they are done away with in this research. In the current research, the PCC is used as a traditional similarity measure to find neighbor users. Another option to find neighbor users would be the use of clustering techniques. The reason behind using clustering algorithms is to place similar users in clusters. Users belonging to the same cluster as the target user are known as

neighbor users. The time required for computation in CF algorithms can be reduced significantly since the number of users in a cluster is significantly less than the total number of users. Moreover, ratings by similar users within a cluster are more pertinent than those of users who are not similar. Therefore, recommendation based on users' ratings may be more accurate.

It has been proved that clustering algorithms outperform similarity measures in terms of finding users similar to the target user. Also clustering algorithms help in dealing with data sparsity and high dimensionality problems. In the current research, KMeans clustering method and the Non-Negative Matrix Factorization model for clustering (NNMF) are used as traditional clustering method. Similarity measures can also be used in the rate prediction process. For instance, the prediction of the rating of user a on item p that also factors the relative proximity of the nearest neighbor N is done through Similarity-based Prediction as shown in Eq. (2). The prediction is made as a weighted average of neighbors having had a rating on p .

$$pr(a, p) = \bar{r}_a + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a,b)} \quad (2)$$

An alternative rate prediction method is the Maximizing Average Satisfaction method or the Average method (Ave_pred), which as shown in Eq. (3) calculates the average of ratings of n neighbor users on item P .

$$pr(a, p) = \frac{1}{n} \sum_{i=1}^n r_{ip} \quad (3)$$

Besides prediction, a common way to recommend items to the target user is through selection of the top N items with the highest prediction value. In this work user based collaborative filtering technique is applied to predict the recommendations of the user.

4. Results and discussions

The benchmark dataset named Movie lens dataset is used to experiment which is provided by www.grouplens.org in different variants w.r.t. number of ratings and number of users. The dataset is collected by grouplens research project team at university of Minnesota for ratings on movies. One variant used is Movielens100k data set that consists of 1,00,000 ratings by 943 users for 1682 items scaling from 1 to 5 where 1 represent low negative and 5 represents high positive opinion of the user.

Table 1. Statistics of movielens100k data set

Maximum Ratings per item	583
Minimum Ratings per item	1
Average Ratings per item	59.453
Maximum Ratings per user	737
Minimum Ratings per user	20
Average Ratings per user	106.045
Sparsity	0.063
Frequency of rating value	1(6110), 2(11370), 3(27145), 4(34174), 5(21201)

The statistics of the movie lens data set is shown in table 1.

In order to evaluate the quality of the recommender, Root Mean Squared Error (RMSE) is used because it can measure the quality of recommendation accurately as shown in Eq. (4). RMSE gives the deviation between actual and predicted ratings, the less RMSE represent high quality of recommendations and if the actual ratings are $\{ar_1, ar_2, \dots, ar_n\}$ and predicted ratings are $\{pr_1, pr_2, \dots, pr_n\}$ then RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (ar_i - pr_i)^2}{n}} \quad (4)$$

The results are analyzed by comparing the performance of the proposed Debiased Tendency Recommender (DBT) with the popular existing model named PLWAP-Mine [29], SVD++ [30], CBF [31], slope one predictor(SOP) [32]. PLWAP-Mine is the technique that uses sequential patters on the usage data with tree representation. SVD++ is the single value decomposition strategy for collaborative filtering approach and Content Based Filtering is a popular approach that does not use any knowledge apart from navigation data. Slope one predictor predicts the rating by using baseline of CF technique which computes the average difference between users rated items. Table 2 shows the values of the RMSE measures for the comparative analysis of the proposed Debiased Tendency recommender model.

Table 2. RMSE value of the various recommenders for Movielens 100k data set

SNo	Method	RMSE
1	PLWAP	0.8452
2	SVD++	0.919
3	CBF	0.8127
4	SOP	0.938
5	DBT	0.7652

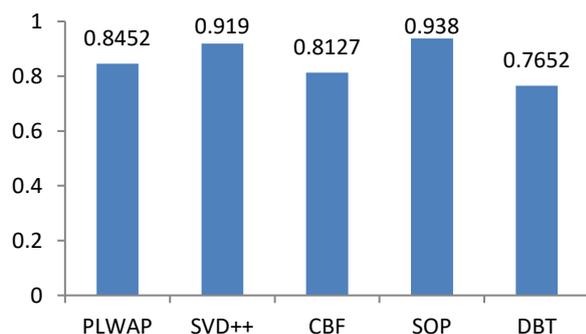


Figure.3 RMSE values of the existing and proposed recommenders for movielens 100K data set

Fig. 3 shows the graphical representation of the results shown in table 2. The graph clearly depicts that the proposed DBR performing better than other popular traditional techniques. The traditional methods directly apply the modeling technique on raw rating information where the ratings distance is far from the centroid of the data. This makes the prediction strategy in turn the error rate to be large. As the proposed DBT strategy is de biasing the rating information personalized to each user, which theoretically reducing the dispersion of the data and in turn reducing the distance from the centroid. Thus the predictions made on de biased rating information obviously leads to low error rate cause less RMSE value when compared to existing strategies.

5. Conclusion

Biased ratings of the items given by users definitely show impact on the quality of the recommendations to the user. De biasing input ratings improves the quality of recommendations. This paper presented a Debaised recommender that debias the rating data then applies collaborative filtering approach to generate recommendations. Through experiments it is observed that the proposed Debaised recommender generating qualitative recommendations when compared with the traditional recommenders. Precision and Recall values are higher for the proposed recommender technique when applied on the benchmark data sets. As future work, the de biasing has to be able to adopt some sophisticated machine learning techniques in debiasing the rating data.

References

- [1] J. Albors, J. C. Ramos, and J. L. Hervas, "New learning network paradigms: Communities of objectives, crowdsourcing, wikis and open source", *International Journal of Information Management*, Vol.28, No.3, pp.194-202, 2008.
- [2] G. Adomavicius, J. Bockstedt, S. Curley and J. Zhang, "Effects of Online Recommendations on Consumers' Willingness to Pay", In: *Proc. of the 6th ACM conference on Recommender Systems*, pp. 40-45, 2012.
- [3] G. Adomavicius, J. Bockstedt, S. Curley, and J. Zhang, "Do Recommender Systems Manipulate Consumer Preferences? A Study of Anchoring Effects", *Information Systems Research*, Vol. 24, No.4, pp.956-975, 2013.
- [4] Q. Yan, J. Ji, and H. Li, "Rating the Raters: Bias Analysis on Yelp Reviews for Improved Star Rating System", *Project report 334, Stanford University*, 2015.
- [5] R. Katarya and O.P. Verma, "An effective collaborative movie recommender system with cuckoo search", *Egyptian Informatics Journal*, Vol. 18, No. 2, pp.105-112, 2017.
- [6] Q. Li, S. H. Myaeng, and B.M. Kim, "A probabilistic music recommender considering user opinions and audio features", *Information Processing & Management*, Vol. 43, No.2, pp. 473–487, 2007.
- [7] S. Cleger-Tamayo, J.M. Fernández-Luna, and J.F. Huete, "Top-N news recommendations in digital newspapers", *Knowledge-Based Systems*, Vol. 27, pp. 180–189, 2012.
- [8] M. Nilashi, O. Ibrahim, N. Ithnin, and N.H. Sarmin, "A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA-ANFIS", *Electronic Commerce Research and Applications*, Vol. 14, No. 6, pp. 542-562, 2015.
- [9] Z. Sun, L. Han, W. Huang, X. Wang, X. Zeng, M. Wang, and H. Yan, "Recommender systems based on social networks", *Journal of Systems and Software*, Vol. 99, pp. 109–119, 2015.
- [10] N. Agarwal, E. Haque, H. Liu, and L. Parsons, "Research Paper Recommender Systems: A Subspace Clustering Approach", *Advances in Web-Age Information Management*, Vol.3739, pp. 475–491, 2005.
- [11] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey", *Knowledge Based Systems*, Vol. 46, pp. 109–132, 2013.
- [12] C.C. Aggarwal, "Recommender systems", *Springer International Publishing Switzerland*, Vol. 40, 2016.
- [13] X. Su and T.M. Khoshgoftaar, "A survey of collaborative filtering techniques", *Advanced Artificial Intelligence*, Vol. 2009, No. 4, 2009.
- [14] L. Cheng and H. Wang, "A fuzzy recommender system based on the integration

- of subjective preferences and objective information”, *Applied Soft Computing*, Vol. 18, pp. 290–301, 2014.
- [15] L. Boratto, “Using Collaborative Filtering to Overcome the Curse of Dimensionality when Clustering Users in a Group Recommender System”, In: *Proc. of the 16th International Conference on Enterprise Information Systems*, pp. 564–572, 2014.
- [16] S. Ghazarian and M.A. Nematbakhsh, “Enhancing memory-based collaborative filtering for group recommender systems”, *Expert Systems with Applications*, Vol. 42, No. 7, pp. 3801–3812, 2015.
- [17] S. Rendle, “Factorization machines”, In: *Proc. of IEEE International Conference on Data Mining*, pp. 995–1000, 2010.
- [18] S. Rendle, “Scaling Factorization Machines to Relational Data”, In: *Proc. of the 39th international conference on Very Large Data Bases*, pp. 337–348, 2013.
- [19] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, “An alternating direction algorithm for matrix completion with nonnegative factors”, *Frontiers of Mathematics in China*, Vol. 7, No. 2, pp. 365–384, 2012.
- [20] X. Zhou, J. He, G. Huang, and Y. Zhang, “SVD-based incremental approaches for recommender systems”, *Journal of Computer and System Sciences*, Vol. 81, No. 4, pp. 717–733, 2015.
- [21] L. Boratto, S. Carta, and G. Fenu, “Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering”, *Future Generation Computer Systems*, Vol. 64, pp. 165–174, 2016.
- [22] M. N. Moreno, S. Segreña, V. F. Lopez, M. D. Muñoz, and A. L. Sanchez, “Web mining based framework for solving usual problems in recommender systems. A case study for movies’ recommendation”, *Neurocomputing*, Vol. 176, pp. 72–80, 2016.
- [23] M. K. Najafabadi, M. N. Mahrin, S. Chuprat, and H.M. Sarkan, “Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data”, *Computers in Human Behavior*, Vol. 67, pp. 113–128, 2017.
- [24] D. Godes and J.C. Silva, “Sequential and temporal dynamics of online opinion”, *Marketing Science*, Vol. 31, No.3, pp. 448–473, 2012.
- [25] F. Nagle and C. Riedl, “Online word of mouth and product quality disagreement”, *Harvard Business School Research Paper*, No. 13-091, 2017.
- [26] L. Muchnik, S. Aral, and S.J. Taylor, “Social influence bias: A randomized experiment. Science”, *American Association for the Advancement of Science*, Vol. 341, No. 6146, pp. 647–651, 2013.
- [27] H. Zhu, B. Huberman, and Y. Luon, “To switch or not to switch: understanding social influence in online choices”, In: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2257–2266, 2012.
- [28] S. Krishnan, J. Patel, M.J. Franklin, and K. Goldberg, “A methodology for learning, analyzing, and mitigating social influence bias in recommender systems”, In: *Proc. of the 8th ACM Conference on Recommender systems*, pp. 137–144, 2014.
- [29] C.I. Ezeife and Y. Lu, “Mining Web Log Sequential Patterns with Position Coded Pre-Order Linked WAP-Tree”, *Data Mining and Knowledge Discovery*, Vol. 10, pp. 5–38, 2003.
- [30] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model”, In: *Proc. of the 14th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining*, pp. 426–434, 2008.
- [31] M. Saveski and A. Mantrach, “Item cold-start recommendations: learning local collective embeddings”, In: *Proc. of ACM Conference on Recommender Systems*, pp. 89–96, 2014.
- [32] D. Lemire and A. Maclachlan, “Slope one predictors for online rating-based collaborative filtering”, In: *Proceedings of SIAM Data Mining*, Vol. 5, pp. 1–5, 2005.