MuSIF: A Product Recommendation System Based on Multi-source Implicit Feedback

Ioannis Schoinas, Christos Tjortjis

MuSIF: A Product Recommendation System Based on Multi-source Implicit Feedback

Ioannis Schoinas[1] and Christos Tjortjis[1*]

[1] School of Science & Technology, International Hellenic University, 14th km Thessaloniki - Moudania
57001 Thermi, Greece

Abstract. Collaborative Filtering (CF) is a well-established method in Recommendation Systems. Recent research focuses on extracting recommendations also based on implicitly gathered information. Implicit Feedback (IF) systems present several new challenges that need to be addressed. This paper reports on MuSIF, a product recommendation system based solely on IF. MuSIF incorporates CF with Matrix Factorization and Association Rule Mining. It implements a hybrid recommendation algorithm in a way that different methods can be used to increase accuracy. In addition, it is equipped with a new method to increase the accuracy of matrix factorization algorithms via initialization of factor vectors, which, as far as we know, is tested for the first time in an implicit model-based CF approach. Moreover, it includes methods for addressing data sparsity, a major issue for many recommendation engines. Evaluation shows that the proposed methodology is promising and can benefit customers and e-shop owners with personalization in real world scenarios

Keywords: Implicit Feedback (IF), Collaborative Filtering (CF), Recommendation Systems (RS), Matrix Factorization (MF), Association Rule Mining (ARM), Artificial Intelligence Applications.

1 Introduction

Content Based Filtering (CBF) systems create customer profiles based on their preferences for various product characteristics or their past behavior [1] [2]. They aim in matching the customer to the items with characteristics closer to their preferences [3]. Collaborative Filtering (CF) systems on the other hand, exploit user ratings to create communities of users or items with similar characteristics, since users with similar tastes tend to prefer the same items [4] [5] [6] [7] [8]. In both cases user input is required. In CBF customers need to create their profiles and explicitly state which attributes they prefer. CF algorithms require user ratings to extract communities.

However, customers often do not create profiles or rate items they preferred or liked. In other cases, system designers might aim to offer a simpler experience to customers, where rating system and preference profiling burdens and makes the process more complex. For such cases explicitly provided information is not available and research has focused in gathering valuable knowledge from Implicit Feedback (IF) [9]. IF is provided by monitoring user behaviour with extensive search and focus is given to purchase history. Though using solely purchase history to create recommendations limits the number of users and items available for recommendations, which can be created just for users who have previously bought any items and only for items that also have been previously bought by someone. This reduces the coverage of a RS, which is an important criterion regarding such systems' performance.

In addition to customer purchases, there are other sources that depict user preferences. Viewing, searching or other domain specific inter-actions can be used to infer customer interests. Purchase history and other implicit sources depict different levels of interest. The act of purchasing shows that the customer was interested enough to purchase this item, though it does not mean that he or she was happy with this choice. In addition, repeated viewing of a specific item shows a definite interest, but not as strong as actually buying it. Thus, different sources should be considered.

Our aim is to adequately infer users' preferences from their behavior. We propose here MuSIF, a Product Recommendation System Based on Multi-source IF. In contrast to most research on IF systems, we utilize different sources of information in addition to purchase history and try to construct an estimate of user preference towards an item [10], [11], [12], [13], [14] [15], [16].The ultimate goal would be to construct an equation that includes a vast number of implicit sources and accurately models user preference. Research exists for systems that use a combination of implicit and explicit feedback [17] [18] [19] or even knowledge derived from users social network [16], though we focus on a purely implicit approach with knowledge derived from user item interactions.

Another issue present in many RS is data sparsity, especially for systems using explicit feedback or systems with a large volume of customers and items. In these cases, customers rate and interact with a small number of items, hindering the performance of recommender systems. Especially in CF systems, it is difficult to recommend items with low to no user interaction; like-wise, is difficult to produce recommendations for customers with very few rates or little interaction history. In our methodology we create a denser user item matrix by enhancing it with the help of Association Rule Mining (ARM). Association Rules (AR) are extracted based on purchase history and fill in missing values where needed.

Aiming to further increase the accuracy of the algorithm, we proceed to apply another optimization technique. The main CF algorithm used is a Matrix Factorization (MF) technique, where we find the latent factor matrices of the user item matrix via optimization. For increasing the performance of our method, we initialize the latent factors prior to using the main algorithm by decomposing the user item matrix with the Singular Value Decomposition (SVD) algorithm. The remaining of the paper discusses related work in section 2, The proposed method and its key components are detailed in section 3. Section 4 presents results and evaluation. Finally, section 5 concludes the paper with directions for further work.

2 Related Work

Research regarding IF is not as popular as explicit recommenders. However, there are a lot of cases where algorithms use IF exclusively or in combination with other types of sources [20]. CF for IF Datasets, which is the backbone of our research, presents an algorithm based solely in IF [10] [11]. The methodology is categorized as CF, and more specifically as MF. The idea is to construct a user item matrix, similar to explicitly created matrix of user-item ratings. Though in this case the matrix consists of estimates of preference induced by the customers views instead of ratings. Preference is set to 1 for each show the user has watched. Notating users with u, items with i, preference with p and the observed times a user watched a show as r, the preference is given by:

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases} \tag{1}$$

The number of times a viewer watched a show depicts their interest towards that show. To model this behaviour confidence is constructed using a predefined constant that sets the increase rate for each view. Notating confidence with c and the predefined constant with a, confidence is given by:

$$c_{ui} = 1 + a\, r_{ui} \tag{2}$$

Factors are extracted minimizing the following cost function:

$$min_{x,y} = \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) \tag{3}$$

In addition, Alternating the Least Squares (ALS) method is used for optimization [21]. The idea is to keep one vector static in each iteration and re-compute the other, lowering the value of the cost function while in the next iteration to alternate the static and re-computed vectors. After computing the user and item vectors, x_u and y_i respectively, the predicted user preference for a specific item is given by the dot product of their factor vectors.

$$\hat{p}_{ui} = x_u^T\, y_i \tag{4}$$

The proposed implementation performs well and sets the ground rules for dealing with IF recommenders. Viewing a show in the specific context is the same as purchasing a product. Repeated purchase of a specific product is a very strong indication of preference. Thus, it is less prone to misinterpretation and more accurate. Though, in the proposed algorithm only a single source of IF is used, which limits recommendations to items that have actually been viewed.

A transformation of the described methodology is presented in [2]. This paper demonstrates different ways for MF and adopts a probabilistic approach. Like the approach in [7], it is concerned with an implicit user item matrix that shows preference between user-item pairs. In this case, the latent factor matrices are calculated using a different cost function and the aim is to maximize it with alternating gradient ascent procedure.

$$p(l_{ui}\,|x_u, y_i, \beta_u, \beta_i) = \frac{\exp(x_i y_i^T + \beta_u + \beta_i)}{1 + \exp(x_i y_i^T + \beta_u + \beta_i)} \tag{5}$$

The research in [12] is another example of IF CF which is also based only on transaction info. Though in this case sequence pattern analysis is included during the construction of the user item matrix, which will be used for the recommendations. The algorithm calculates a preference score derived from transactions, and a preference score derived from sequence pattern analysis on transactions. The aggregation of both scores forms the final preference score of the user item pairs. The preference score using transactions is calculated with the following equations:

$$AP_{ui} = \ln\left(\frac{t_{ui}}{T_u} + 1\right) \tag{6}$$

$$RP_{ui} = \frac{AP_{ui}}{\max_{c \in U}(AP_{ci})} \tag{7}$$

$$IR_{ui} = round\ (5\ RP_{ui}) \tag{8}$$

Then the system proceeds with the prediction of the user preference using ratings from the k most similar users. The prediction is computed with:

$$CFPP_{ai} = \overline{R_a} + \frac{1}{\sum_{b=1}^{k} |\,sim(a,b)|} \sum_{b=1}^{k} sim(a,b)(R_{bi} - \overline{R_b}) \tag{9}$$

For the sequential pattern analysis, the algorithm creates sequences based on the transaction of all the other customers. Then it proceeds with comparing the derived sequences with the transaction's subsequences of the target customer. Finally, the score derived from the analysis is found from the equation:

$$SPAPP_{ai} = \sum_{s \in SUB} Support_s^i \tag{10}$$

In this equation, the set of the user's subsequences is denoted as SUB, and the support of the item in the specific subsequence as Support s. The final preference score FPP for the user item pair is calculated using the normalized CFPP and SPAPP and the constant a as weight in the following manner:

$$FPP_{ui} = a\ NCFPP_{ui} + (1 - a)NSPAPP_{ui} \tag{11}$$

During evaluation the algorithm was tested against CF and sequential pattern analysis with regards to precision, recall and F1. The results show that the use of implicit rating is an acceptable solution to insufficient explicit feedback. In this example there is a combination of IF CF algorithm with sequential pattern analysis. Sequential pattern and market basket analysis are well established research topics in data mining regarding transactions. Since IF datasets contain information regarding transactions, they make such techniques important tools for increasing accuracy and performance.

Another interesting approach aiming at increasing the performance of MF techniques is via initialization of factor matrices. In CF with initialized factor matrices [18] offers a methodology to improve approaches such as the one used in our implementation and those in papers [10], [11] where they randomly initialize the user item factor matrices and implement a repeated optimization method on these matrices. In these cases, random initialization is not the optimal method.

In the first phase of the methodology, latent user and item factor matrices are extracted from the user-item matrix. The initialization method consists of the following steps. First additional values are being filled in the original user-item matrices, where the rating is unknown. This is accomplished by using a method such as averaging the known user ratings for the specific item. Eight approaches regarding filling the unknown rating values prior to SVD have been tested. These include using the median rating of user, the median rating of item, the total median of items, the average of user and item median, the average ratings of the user, the average ratings of the item, total average of all ratings and the average from user and item averages.

For evaluation the proposed approach was tested against a randomly initialized matrices MF algorithm with the same optimization method. Results show that the method improved the system, while requiring less iterations to find the matrices with the least error. This led to increased execution time, even double time in some cases. The approach describes a method of increasing accuracy and overall performance for discovering latent factor models. What should be noted though is that taking average of ratings for filling the user-item matrix before SVD requires further examination and testing over other possible alternatives that could further increase performance.

3 Method

Our methodology is a combination of three main approaches. The main algorithm we use is a variation of the IF CF using ALS. In a similar fashion, we construct a user item matrix of inferred preference induced by monitoring customer behavior. Though in our case we utilize three different sources of implicit feedback. In addition, we combine the algorithm with ARM to address the data sparsity issue present in our dataset. In the third part we aim at further increasing performance by initializing user and item factor vectors

3.1 Implicit Matrix Factorization (MF)

The main algorithm of our system is an implementation based on the work of CF for IF dataset [7, 13]. For the implementation of the algorithm, the raw data need to be transformed in a user-item matrix with values representing the implicit rating associated with the pair. Our dataset provides observations over different visitors' behavior. Thus, the first stage of the algorithm involves preprocessing raw data to transform them in a user-item matrix.

In CF for IF Datasets a user's interest towards a show is inferred by the number of times they viewed the show. In our case instead of a single implicit source we consider three. Thus, interaction score includes three different observations and is given by:

$$r_{ui} = v_{ui} + b_{ui} + t_{ui} \tag{12}$$

The number of times the visitor viewed the item is notated as v, the number of times the visitor added the item to their cart is notated as b, and the number of times the visitor bought the item as t.

During the purchase process the visitor usually navigates through several items before deciding. While viewing products, it is possible that one sees products that they will then exclude for several reasons; one being that they did not like them. Viewing items, even though being an indication that the visitor wanted to learn more about them, does not always mean that the customer prefers it. On the other hand, “add to cart” and transaction are stronger indications of preference, since the visitor did decide to purchase them. Thus, the three observations have different weight regarding visitor’s preference over an item. Taking into consideration the above weighting of the observations is required, thus the interaction score is transformed into:

$$r_{ui} = v_{ui} + b_{weight} * b_{ui} + t_{weight} * t_{ui} \tag{13}$$

As t_{weight} we denote the transaction weight and as b_{weight} the “add to cart” weight. After cross validation we defined transaction weight to be equal to 100, while “add to cart” to be 50. We considered not giving additional weight to “add to cart” since the customer is required to proceed to it if he wants to buy it, and this intent is captured by the “transaction” interaction. Finally, “add to cart” was included because there might be cases where the customer added the item to the cart, but did not complete the transaction. This is also a strong indication of preference, though not as strong as actually buying it.

In the proposed algorithm, preference is a binary variable which takes the value of 1 for the cases where a user interacted with an item and 0 for the rest. With r_{ui} we notate the interaction score for a user u with an item i.

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases} \tag{14}$$

Also, in our case, it is needed to consider differences in preference level and so a confidence level is defined. Similarly, repetition of interaction is considered as a strong indication of preference. Confidence of each user item pair is given by:

$$c_{ui} = 1 + a(\, r_{ui}\,) \tag{15}$$

A predefined constant used as scale regulator is notated as a. 40 was found after validation to perform well in our model. After creating the user item matrix, the next phase of the algorithm is to calculate the user item factor vectors. The system proceeds with calculating the vectors by minimizing the cost function proposed in [7]:

$$\min_{x,y} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2\right) \tag{16}$$

As x is denoted the user factor vector and as y the item factor vector. The $\lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2\right)$is used to prevent overfitting. After having calculated the factor vectors, we can get a prediction for the user-item pair by calculating the dot product of their factor vectors.

$$rec_{ui} = x_u^T \cdot y_i \text{rec}_{\text{ui}} \tag{17}$$

is the predicted preference score for a specific user towards a specific item. The system proceeds with ranking all products for a user, and a list ordered by preference can be extracted and used as recommendations.

3.2 Association Rule Mining (ARM)

AR in our system are used for enhancing the CF algorithm and addressing the data sparsity problem. The main algorithm calculates the user item factor vectors based on the user item matrix. The denser the matrix is the more accurate the results become. Similarly to [5] ARM is used in order to make the user-item preference matrix denser by extracting rules and filling missing values accordingly. We used AR extracted by Apriori considering all the transactions for each user [22]. The rules provide us with information about items that tend to be bought together.

The system fills the matrix with specific user items, where the user has bought the base item, but not the appended item. The score assigned to the previously unassigned user item value takes is derived from the score of the base item. In this way we can also model cases where the user bought the base item more than once. In addition, since confidence provides information about the correlation between base and appended items, this should also be considered in calculating missing values. Thus, the score is equal to that of base item times the confidence of the rule. Notating user with u, items with i, i_a are the appended items and i_b are the base items and nr is a newly assigned rating score that was previously absent for the specific user item pair.

$$nr_{ui_a} = r_{ui_b} * confidence \tag{18}$$

After enhancing the user-item score the system proceeds by discovering the user item factor vectors of the enhanced matrix and the remaining phases of the recommendation system.

3.3 Initializing user item factor vectors

In the last phase of our approach, we aim at further increasing the performance of the system by initializing the factor vector prior to ALS optimization. The methodology presented in [10], [11] uses random values for initialization of the vectors. Following the research in [18] we proceed by decomposing the initial user item matrix using the SVD algorithm. In contrast to this method, we did not proceed by filling the missing values with averages. Average filling technique is a stochastic approach and further research is required to discover a more suitable technique. A flowchart for our approach is shown in Fig. 1.

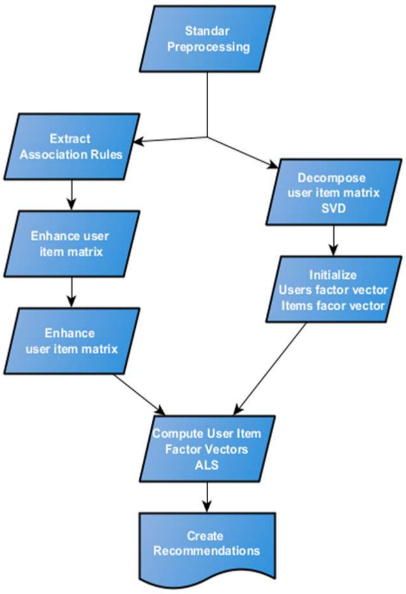

Fig. 1. MuSIF's Flowchart

4 Evaluation

For the evaluation of the system we tested the algorithm on various transformations of the user-item matrix. Since negative feedback for the recommended items was not in the scope of our work, accuracy metrics are not suitable for the evaluation of the system. Thus, the metric we used for evaluating the quality of the proposed methodology is the Mean Percentile Ranking (MPR).

The first thing that needs to be evaluated is if the algorithm performs better considering all event types or considering only transaction events. Thus, two user item matrices are constructed using the proposed methodology, one constructed based solely on transactions and one including all three interactions. We refer to the user-item matrix that includes all interactions as multi-type MT matrix, and to the one that contains only transactions as single-source matrix ST.

Furthermore, the algorithm was evaluated on how well it performs when applying methods for increased accuracy. Thus, the tests on each of the previous user-item matrices implement four different methods: i) the standard method, ii) enhanced by use of ARM, iii) by initializing the user item factor vectors method and iv) by enhancing the matrix as well as initializing the factor vectors method. With 4 different methods on two user item matrices, we calculated the mean percentile ranking for eight different cases.

For AR enhancing the user level category, extracted rules were used as described section 3.2. In total 23 rules have been extracted using minimum confidence of 0.5. Implementing changes where it was necessary lead to 44 changes in the initial user-

item matrix. For initialization of the factor vectors the SVD decomposed the initial user-item matrix without any prior filling of missing values.
In addition, on both matrices, information about visitors who have viewed less than 5 items was deemed as unusable and was removed from the training set, since the it is insufficient to produce meaningful recommendations. In addition, this serves the purpose of reducing the total size of the evaluated training set and reducing its sparsity. Even though this hinders the system's recommendation coverage, addressing the cold start issue was out of scope of this work [23].
The dataset used for evaluation is the Retailrocket recommender system dataset [24]. Retail Rocket is a company that develops personalization technologies for their customers. The dataset consists of three different files each of which is associated with a different aspect of an e-commerce system.

4.1 Single-source evaluation with transaction events

Table 1. Single source methods notation

Method	Notation	Score
single -source	ST	30
enhanced single -source	EST	38
initialized single -source	IST	39
enhanced & initialized single -source	EIST	39

In this case the standard methodology provides the best results. Unexpectedly, all the methods applied aiming at increasing system accuracy turned out to greatly hindered it instead. The assumption is that in this case, the user-item matrix uses a small number of users and items and is dense enough for the algorithm to perform well, in which case any attempt to add values or initialize the factor vectors adds noise to the matrix.

4.2 Multi-source evaluation with transactions, views and addtocart events

Table 2. Multi source methods notation

Method	Notation	Score
multi-source	MT	38.31
enhanced multi-source	EMT	35.18
initialized multi-source	IMT	36.37
enhanced & initialized multi-source	EIMT	37.64

In this case all the applied methods for increased accuracy actually did increase it. Notably AR enhancing introduced the largest improvement of 3.13%. Following AR initialization of the factor vectors improved the standard methodology by 1.94%. Though it was assumed that implementing both enhancement and initialization would improve performance even further, though that is not the case. Combination of both improvement methodologies improved the algorithm by a mere 0.67%.

4.3 Single-source vs multi-source comparison

By comparing the results considering single-source and multi-source matrices, it is notable that single-source user-item matrix performs a lot better than any other type of matrix evaluated. Specifically, compared to the best performing case of multi-source matrices, the AR enhanced multi-source matrix, still performed 5.18% better.
The reason that the single-source performs better is not exclusively because it is a better model. The main difference in the two user-item matrices is data sparsity. Sparsity greatly affects MF techniques. In both matrices, users with only a few interactions have been eliminated. This resulted in a denser single-source matrix. More specifically, the multi-source matrix has a sparsity degree of 0.01%, which means that only the 0.01% of user-item pairs have ratings. On the other hand, single-source matrix has a sparsity degree of 1%, which means that only 1% of all possible user-item ratings are filled. Even though single-source matrix is also too sparse, it is ten times denser than the multi-source matrix.

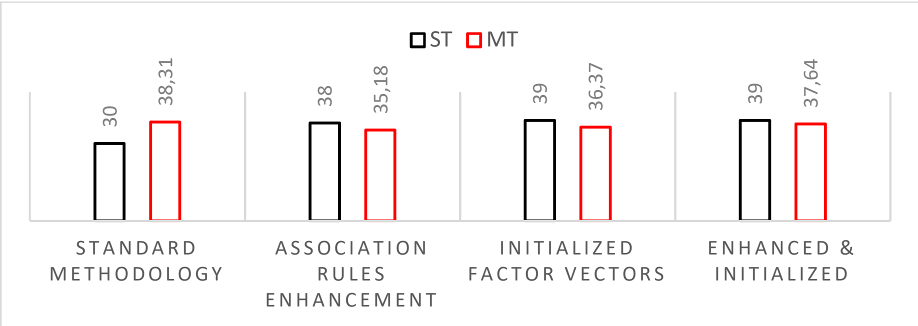

Fig. 2. Comparison of single-source and multi-source algorithms on all methods

4.4 Coverage

In contrast to accuracy, a quality measure often required in RS is coverage. In terms of coverage, the single-source matrix can create recommendations for 1056 users on 106 different items. In contrast, the multi-source matrix can recommend 16022 different items to 43827 users. In terms of coverage the multi-source matrix is more suitable for creating recommendations.

5 Conclusions and Future Work

In this paper we proposed MuSIF, a Product Recommendation System based on Multi-source Implicit Feedback. MuSIF utilises a variation of the IF CF by means of ALS. It constructs a user item matrix of inferred preference induced by monitoring customer behavior and uses three different sources of IF. We proposed to treat each source in accordance with the level of interest it implies. In addition, it incorporates ARM to address the data sparsity issue and further increases performance by initializing user and item factor vectors. Conclusions can be categorized regarding the multi

or single-source of implicit information used, the AR extract-ed, the method to enhance the user-item matrix with AR, and the method to initialize factor vectors.
Regarding the use of different types of implicit information, results show that a lot of noise is added hindering accuracy. Though predictions are far from random, thus, considering also the difference in coverage between single-source and multi-source (14966 more users and 15916 more items in multi-source) deems multi-source a promising approach. The multi-source approach requires further research for discovering a more suitable method to include this information.
Regarding ARM, even though the number of extracted rules is very small for adequately addressing data sparsity on their own, results indicate that they benefit matrices with data noise, such as multi-source matrices. Initialization of the factor vectors, like AR enhancement, benefits matrices with noise and can improve their accuracy. Though further research is required regarding the decomposition of the matrix as to discovering the best approach for filling the user-item matrix from which the initial vectors will be extracted.
Future research should examine different ways of including different implicit information in the user-item matrix and best performing models. The literature for predicting future transactions proposes treating other implicit information as auxiliary data and implements a regression model for discovering the impact they pose in the final transaction [18]. Implicit datasets that offer transaction history provide opportunities for sequence analysis and rule-based recommenders in general. In this paper we considered only AR and let sequential pattern analysis as one of the best next steps for improving the method.
Factor vectors initialization should be further researched for accuracy improvement. Specifically, an adequate way of filling the initial matrix prior to decomposition should be discovered. The literature proposes averages [18] which improved accuracy. In addition, other ways for singular value decomposition should also be examined. The data sparsity issue should also be examined further. ARM alone did not manage to increase the matrices significantly. One possible way could be via product similarity based on item properties and using Content-based filtering methods. Lastly, a lot of products are suitable only for specific time periods such as seasons or a specific age. Thus, time should also be considered in product recommendation research. Time depended recommenders are focused ov by some researchers [25].

References

1. M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems", in *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer, pp. 325-341, 2007.
2. V. C. Gerogiannis, A. Karageorgos, C. Tjortjis and L. Liu, "Personalised Fuzzy Recommendation for High Involvement Products", *IEEE Int'l Conf. Systems, Man, and Cybernetics* (SMC 2013), pp. 4884-4890, 2013,
3. F. Ricci, L. Rokach, B. Shapira and P. Kantor, *Recommender Systems Handbook*, 2011.
4. S. Sivapalan, A. Sadeghian, H. Rahanam and A. Madni, *Recommender Systems in E-Commerce,* 2014.

5. J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative Filtering Recommender Systems", in *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 291-324, 2007.
6. G. Linden, B. Smith and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering", *IEEE Internet Computing*, vol. 7, pp. 76-80, 2003.
7. J. S. Breese, D. Heckerman and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", in *Proc. 14th Conf. Uncertainty in AI*, pp. 43-52, 1998.
8. S. Pathan, K. Panjwani, N. Yadav, S. Lokhande and B. Thakare, "Product Recommendations System Survey", *Int'l Journal of Computer Applications*, Vols. 131-9, p. 3, 2015.
9. G. Jawaheer, M. Szomszor and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service", *Proc. 1st Int'l Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pp. 47-51, 2010.
10. Y. Hu, Y. Koren and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets", in *Proc. IEEE Int'l Conf. on Data Mining*, pp. 263-272, 2008.
11. C. C. Johnson, *Logistic Matrix Factorization for Implicit Feedback Data*, 2014.
12. K. Choi, D. Yoo, G. Kim and Y. Suh, "A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis", *Electronic Commerce Research and Applications*, vol. 11, pp. 309-317, 2012.
13. L. Baltrunas and X. Amatriain, "Towards Time-Dependant Recommendation based on Implicit Feedback", *Workshop on context-aware recommender systems*, 2009.
14. S. A. Bahrainian, S. M. Bahrainian, M. Salarinasab and A. Dengel, "Implementation of an Intelligent Product Recommender System in an e-Store", *Int'l Conf. Active Media Technology, AMT 2010*, pp 174-182, 2010.
15. B. Pradel, S. Sean, J. Delporte, S. Guerif, C. Rouveirol, N. Usunier, F. S. Fogelman and F. Dufau-Joël, "A case study in a recommender system based on purchase data", in *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 377-385, 2011.
16. J. Delporte, A. Karatzoglou, T. Matuszczyk and S. Canu, "Socially Enabled Preference Learning from Implicit Feedback Data", *ECML PKDD: Machine Learning and Knowledge Discovery in Databases*, pp 145-160, 2013.
17. T. Gábor, P. István and T. Domonkos, "Applications of the Conjugate Gradient Method for Implicit Feedback Collaborative Filtering", *Proc. 5th ACM conf. on Recommender systems*, pp. 297-300, 2011.
18. G. Guo, H. Qiu, Z. Tan, Y. Liu, J. Ma and X. Wang, "Resolving Data Sparsity by Multi-type Auxiliary Implicit Feedback for Recommender Systems", *Knowledge-Based Systems*, vol. 138, pp. 202-207, 2017.
19. D. Z. a. D. T. I. Pilászy, "Fast ALS-based matrix factorization for explicit and implicit feedback datasets", in *Proc. 4th ACM Conf. Recommender Systems*, pp. 71-78, 2010.
20. R. Dutta and D. Mukhopadhyay, "Offering A Product Recommendation System in E-commerce", 2011.
21. Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems", *Computer*, vol. 42, no. 08, pp. 30-37, 2009.
22. R. Agarwal and R. Srikant, "Fast algorithms for mining association rules", in *Proc. of the 20th VLDB Conf.*, pp. 487-499, 1994.
23. O. Nalmpantis and C. Tjortjis, "The 50/50 Recommender: a Method Incorporating Personality into Movie Recommender Systems", *CCIS Communications in Computer and Information Science*, pp. 498-507, 2017.
24. Retailrocket recommender system data www.kaggle.com/retailrocket/ecommerce-dataset
25. P. Jiang, Y. Zhu, Y. Zhang and Q. Yuan, "Life-stage Prediction for Product Recommendation in E-commerce", in *21th ACM SIGKDD Int'l Conf.*, pp. 1879-1888, 2015.