

Research Article

The High Security Mechanisms Algorithm of Similarity Metrics for Wireless and Mobile Networking

Xingwang Wang^{1,2}

¹*School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China*

²*Shanghai Vocational Technical College of Agriculture & Forestry, Shanghai 201699, China*

Correspondence should be addressed to Xingwang Wang; wangx.w@shu.edu.cn

Received 9 March 2017; Accepted 14 May 2017; Published 20 July 2017

Academic Editor: Arun K. Sangaiah

Copyright © 2017 Xingwang Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of human society and the development of Internet of things, wireless and mobile networking have been applied to every field of scientific research and social production. In this scenario, security and privacy have become the decisive factors. The traditional safety mechanisms give criminals an opportunity to exploit. Association rules are an important topic in data mining, and they have a broad application prospect in wireless and mobile networking as they can discover interesting correlations between items hidden in a large number of data. Apriori, the most influential algorithm of association rules mining, needs to scan a database many times, and the efficiency is low when the database is huge. To solve the security mechanisms problem and improve the efficiency, this paper proposes a new algorithm. The new algorithm scans the database only one time and the scale of data to deal with is getting smaller and smaller with the algorithm running. Experiment results show that the new algorithm can efficiently discover useful association rules when applied to data.

1. Introduction

With the rapid development of web technology, the number of choices is becoming overwhelming. It takes a long time to filter, prioritize, and efficiently deliver relevant information so as to alleviate the problem of information overload. Recommender systems [1] have grown so fast that they can meet the needs of users' ambiguous requirements. They utilize statistic method and knowledge discovery technology, providing users with personalized content and services by searching through large volume of dynamically generated information. Recently, various approaches for building recommender systems have been developed, which can utilize collaborative filtering, content-based filtering, or hybrid filtering [2–4]. Among the above filtering techniques, collaborative filtering recommendation is the most mature and the most commonly implemented. Collaborative filtering technique can be divided into two classifications; they are model-based filtering and memory-based filtering. The model-based filtering learns a model from the user-item ratings which can be computed offline. Once the model is generated, the

process of prediction will be easy and fast. Lots of model-based filtering techniques have been proposed by researchers such as Latent Semantic Indexing (LSI) [5], decision tree [6], Bayesian network approach models [7], and cluster models [8, 9]. Usually, model-based algorithm has better scalability but lower accuracy than memory-based algorithm. Although collaborative filtering technique is commonly used, it still encounters one crucial issue remaining to be solved, namely, data sparsity problem [10–13], thus leading to the nonoptimal nearest neighbors because the core of the collaborative filtering algorithm is to find the k -nearest neighbors [14–18]. For lack of reference rating values, this step of searching neighbors causes big inaccuracy. In the traditional collaborative filtering algorithm, such similarity metrics are used to calculate the similarity between users or items as cosine, Pearson-correlation, and modified cosine [19–22]. All of them present poor performance when they are applied to big data with high sparsity. This paper proposes a new algorithm, considering both user similarity and item ones. Matrix prefilling, a method of preprocessing, is based on association rules which are not proposed by others before

when measuring similarity. Experimental results of the proposed model on a real dataset: the dataset proves to generate more accurate prediction results compared to the traditional ones. The remainder of this paper is organized as follows. Section 2 is a brief introduction of association rule whose concept and algorithm will be used in Section 3 to propose a new algorithm. Section 3 focuses on the algorithm for wireless and mobile networking which is the highlight of this paper. Experimental results and analyses are displayed in Section 4. Section 5 is the final part of this paper in which conclusion is reached.

2. Related Work

2.1. Related Concepts of Association Rules. Transaction database $D = \{t_1, t_2, \dots, t_k, \dots, t_n\}$ is the set of all the transactions. $I = \{i_1, i_2, \dots, i_m\}$ is the set of all the items in D [23–25]. Every transaction contains a set of items which is the subset of I . Item set is a collection that contains 0 or more items. If the number of items an item set contains is k , then the item set is called k -item set. *Support* count is an important property of an item set. It indicates the number of a particular item set contained in the transactions. $\sigma(X)$, the *support* count of item set X , is defined as follows:

$$\sigma X = |\{t_i \mid X \subseteq t_i, t_i \in D\}|. \quad (1)$$

$|\dots|$ represents the number of elements in the collection.

A rule is defined as an implication form $X \rightarrow Y$, where $X \subset I, Y \subset I$, and $X \cap Y = \phi$. *Support* and *confidence* are two important measures of association rules. *Support* indicates the frequency of the rule in a dataset. It is defined as

$$\text{support}(X \rightarrow Y) = \frac{\sigma X \cup Y}{N}. \quad (2)$$

N is the total number of transactions.

The *confidence* of a rule $X \rightarrow Y$ is the proportion of transactions that contains X which also contains Y . It is defined as

$$\text{confidence}(X \rightarrow Y) = \frac{\sigma X \cup Y}{\sigma X}. \quad (3)$$

Support and *confidence* are two important measures to evaluate association rules. Rules with low *support* may occur only occasionally which are meaningless in most cases. Therefore, *support* is often used to delete those meaningless rules. *Confidence* is a measure of accuracy of association rules. If the *confidence* of the rule $X \rightarrow Y$ is high, the possibility of Y appearing in the transactions which contain X is larger.

2.2. Apriori Algorithm. Apriori is a typical algorithm with candidate set generated. It uses the *support* based pruning method and a level-wise and breadth-first search to discover the frequent item sets. Apriori uses two properties below to compress search space.

Lemma 1. *If the item set X is frequent, then all nonempty subsets are frequent too.*

Lemma 2. *If the item set X is nonfrequent, then all supersets are nonfrequent too.*

Candidate item set generation is a very critical step. It should ensure that the candidate item sets are complete while avoiding too many unnecessary candidates. This step consists of two parts.

(1) In the join step, this paper joins two frequent $(k-1)$ -item sets $L1$ and $L2$ to generate candidate k -item sets. This paper should make sure that the first $k-2$ items of $L1$ and $L2$ are the same. Then, the first $k-2$ items and the last item of $L1$ as well as the last item of $L2$ compose the candidate k -item set.

(2) In the pruning step, this paper uses a strategy to delete some unnecessary candidates. According to Lemmas 1 and 2, for each k -item set generated, this paper examines whether all the $k-1$ subsets are frequent. If not, this paper removes it from the candidate k -item sets.

Apriori algorithm effectively filters the unnecessary candidates. It will get a good data mining result, especially for short pattern data. However, the weakness is that the database needs to be scanned many times. It will produce tremendous I/O cost. Another weakness is that a lot of candidate item sets may be generated. It will cost a lot of time and memory space.

2.3. FP-Growth Algorithm. FP-Growth is a classic algorithm without candidate item sets generated. It compresses the data into a structure called FP-tree. The frequent item sets are discovered by doing a recursive search of the FP-tree.

The process of FP-Growth mainly consists of two steps.

(1) *Constructing the FP-Tree.* When the database is scanned for the first time, this paper selects the items which satisfy the minimum *support* and puts these items to a header table with a descending sort order according to *support*. When the database is scanned for the second time, the items contained in a transaction are sorted according to their order in the header table and are inserted in the FP-tree. Then combine the same paths in the tree.

(2) *Discovering Frequent Item Sets by Searching the FP-Tree.* If the FP-tree contains only one path, enumerate all the possible item sets. If not, for each item in the header table, this paper creates its conditional pattern base so as to construct the conditional pattern tree. The recursive process will not stop until the tree is empty.

FP-growth algorithm scans the database only two times and avoids the generation of candidate item sets, but the weakness is that when the database is huge, the FP-tree is too large and even cannot be constructed in memory because all the records in database are compressed into the FP-tree.

3. The Improved Apriori Algorithm Based on Matrix

To avoid the weakness of apriori algorithm, this paper proposes an improved algorithm on the basis of apriori algorithm. This paper converts the transaction database to

TABLE 1: A transaction database.

TID	Items
1	I_2, I_5
2	I_1, I_2, I_4
3	I_1, I_3, I_4
4	I_2, I_3, I_4, I_5

TABLE 2: A Boolean matrix.

R	I_1	I_2	I_3	I_4	I_5
T_1	0	1	0	0	1
T_2	1	1	0	1	0
T_3	1	0	1	1	0
T_4	0	1	1	1	1

a Boolean matrix and deletes the unnecessary rows and columns of the matrix to reduce the scale of the data.

3.1. Related Concept. Association rules usually focus on transaction databases. If this paper converts the transaction database to a Boolean matrix, on the one hand, the database can be scanned only one time so as to reduce the cost of I/O and, on the other hand, it may reduce the memory consumption when the data is in the form of 0 and 1.

Definition 3. Let $I = \{I_1, I_2, \dots, I_n\}$ be an item set and $T = \{T_1, T_2, \dots, T_m\}$ be a set of transactions in the database and each transaction in T has a unique transaction id called TID. The method by which transactions are converted into a Boolean matrix is as follows: let R be the binary relation from I to T . $r_{ij} = R(T_i, I_j)$, $R = (r_{ij})_{m \times n}$. Then

$$r_{ij} = \begin{cases} 1, & I_j \in T_i \\ 0, & I_j \notin T_i \end{cases} \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n. \quad (4)$$

An example of a transaction database is in Table 1. The Boolean matrix of the database is in Table 2.

The column vector I_j of the Boolean matrix is defined as $I_j = \{r_{1j}, r_{2j}, \dots, r_{mj}\}$. The *support count* of I_j is

$$\text{support_count} \{I_j\} = \sum_{i=1}^m (r_{ij}). \quad (5)$$

For k -item set $\{I_1, I_2, \dots, I_k\}$, its *support count* is

$$\begin{aligned} &\text{support_count} \{I_1, I_2, \dots, I_k\} \\ &= \sum_{i=1}^m (r_{i1} \wedge r_{i2} \wedge \dots \wedge r_{ik}). \end{aligned} \quad (6)$$

\wedge is "and" operation. When I_1, I_2, \dots, I_k are simultaneously 1, the *support count* is incremented by 1.

Lemma 4. *If the number of "1" instances contained in a row of Boolean matrix is less than k , then when this paper counts the support of k -item set, this row can be deleted from the matrix.*

According to the definition of support count,

$$\begin{aligned} &\text{support_count} \{I_1, I_2, \dots, I_k\} \\ &= \sum_{i=1}^m (r_{i1} \wedge r_{i2} \wedge \dots \wedge r_{ik}). \end{aligned} \quad (7)$$

If the number of "1" instances contained in a row is less than k , there will exist j which makes $r_{ij} = 0$; then $r_{i1} \wedge r_{i2} \wedge \dots \wedge r_{ik} = 0$. Therefore this row makes no contribution to the support count of k -item set.

Lemma 5. *If there is an item I_j , the number of I_j instances that appear in frequent k -item sets L_k is less than k ; the column of I_j can be deleted in the process of frequent $(k + 1)$ -item set generation.*

Let Y be a frequent $(k + 1)$ -item set; then all its k -subsets are frequent. For each $I_j \in Y$, the number of I_j instances that appear in frequent k -item sets should be k . If the number is less than k , then I_j will not be the element of the frequent $(k + 1)$ -item set.

3.2. The Searching of k -Nearest Neighbors. After the process above, this paper takes user similarity into account. The similarity of user i and user j is computed as (8). I denotes the set of all the items.

$$\text{sim}(i, j) = \frac{\sum_{c \in I} (R_{i,c} - \bar{M}_i) (R_{j,c} - \bar{R}_j)}{\sqrt{\sum_{c \in I} (R_{i,c} - \bar{R}_i)^2} \sqrt{\sum_{c \in I} (R_{j,c} - \bar{R}_j)^2}}. \quad (8)$$

For each user u , the aim to find the k -nearest neighbor is to find a user set $U = \{U_1, U_2, \dots, U_k\}$, $u \notin U$, $\text{sim}(u, U_1)$ has the highest value, and $\text{sim}(u, U_2)$ has the second highest value, and so on.

3.3. The Generation of Recommendation. After the step of finding the k -nearest neighbors, the next step is to generate recommendations. Let the set of k -nearest neighbors of user u be NN_u and the rating that user u give to the item i be $R_{u,i}$; the calculation is as follows:

$$\begin{aligned} &R_{u,i} \\ &= \begin{cases} \bar{R}_u + \frac{\sum_{n \in \text{NN}_u} \text{sim}(u, n) \times (R_{n,i} - \bar{R}_n)}{\sum_{n \in \text{NN}_u} (|\text{sim}(u, n)|)}, & \text{if } r_{u,i} \cdot \text{flag} = 2 \\ r_{u,i}, & \text{if } r_{u,i} \cdot \text{flag} = 0. \end{cases} \end{aligned} \quad (9)$$

3.4. Description of the Improved Algorithm. The process of the improved algorithm is described in Algorithm 1. First this paper converts a database to a Boolean matrix. Then according to Lemmas 4 and 5 the unnecessary rows and columns of the matrix are deleted with the algorithm running.

The improved algorithm based on matrix is shown in Algorithm 1.

3.5. Evaluation Criteria. Not all association rules are useful, so it is necessary to select the association rules in which we

Input: dataset D , minimum *support* minsup
Output: all the itemsets satisfied with minsup

- (1) Scan transaction database D and convert it to a Boolean matrix.
- (2) Calculate the *support* of every 1-itemset. Item sets whose *support* are not less than min sup compose frequent 1-item sets L_1 . Delete the columns of the infrequent items. Delete the rows in which the number of "1" contained is less than 2.
- (3) for ($k = 2; L_{k-1} \neq \phi; k++$) do begin
- (4) Combine the items of each column and generate candidate k - item sets C_k .
- (5) Calculate the *support* of C_k .
- (6) Item sets whose *support* is not less than min sup compose frequent k -item sets L_k .
- (7) Delete the columns of the Items which are contained in infrequent item sets and the number Of which appear in L_k is less than k .
- (8) Delete the rows in which the number of "1" contained is less than $k + 1$
- (9) End

ALGORITHM 1: The procedure of the improved algorithm based on matrix.

are interested. *Support* and *confidence* are two basic criteria to evaluate if an association rule is useful. However in some case the two criteria may give us an unexpected suggestion. So this paper uses the criterion called *lift* to evaluate the association rules in addition to *support* and *confidence*. The *lift* of a rule $X \rightarrow Y$ is defined as

$$\text{lift}(X \rightarrow Y) = \frac{\text{confidence}(X \rightarrow Y)}{\text{support}(Y)}. \quad (10)$$

Lift is the ratio of a rule's *confidence* and the consequent's *support*. If the value of *lift* is 1, X and Y are independent. If the value is above 1, X and Y are positively correlated. If the value is below 1, X and Y are negatively correlated.

3.6. Performance Analysis. Compared with apriori algorithm, the improved algorithm scans the database only one time. It converts the transaction database to a matrix. The remaining steps are operated on the matrix without scanning the database again. This will reduce the I/O cost. The other advantage of the improved algorithm is that the scale of data to be dealt with is getting smaller and smaller with the algorithm running. In the process of frequent item sets generation, the columns of items which will not be contained in frequent item sets and the rows which make no contribution to the *support* count will be deleted. Therefore the scale of the matrix will be smaller and smaller and the efficiency will be improved a lot. On the other hand, when a transaction contains many items, compared with transaction list, a Boolean matrix occupies less memory space.

4. Results and Analysis

To access the performance of the improved algorithm, this paper uses apriori algorithm and the improved algorithm proposed in this paper to mine frequent item sets from different agricultural databases. The experiments were performed on an Intel i5-2450 processor 2.5 GHz with 4G memory, running Windows 8. This paper used R language to code the algorithms.

TABLE 3: Runtime of the two algorithms on mushroom dataset.

<i>Support</i>	0.60	0.65	0.70	0.75	0.80
Apriori algorithm	3.25	2.12	1.95	1.85	1.72
Improved algorithm	2.72	1.73	1.63	1.54	1.42

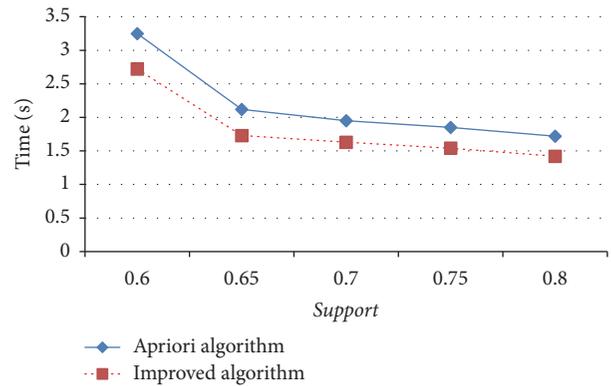


FIGURE 1: Runtime of the two algorithms on mushroom dataset.

Table 3 and Figure 1 show the performance of the two algorithms in the UCI dataset named mushroom. The dataset contains 7847 records and 118 items. The minimum *confidence* is set to be 0.5 and the minimum *support* is set, respectively, to be 0.60, 0.65, 0.70, 0.75, and 0.80.

Table 4 and Figure 2 show the performance of the two algorithms in the UCI dataset named soybean. The dataset contains 5264 records and 655 items. The minimum *confidence* is set to be 0.5 and the minimum *support* is set, respectively, to be 0.75, 0.76, 0.77, 0.78, 0.79, and 0.80.

The results show that the runtime of the improved algorithm is less than apriori algorithm. The improved algorithm is more effective than apriori algorithm.

The evaluation method *lift* is used to optimize the mining result. A subset of the mining result of mushroom dataset is shown in Algorithm 2. Algorithm 2 shows the association rules whose *support* and *confidence* are high but whose *lift* is 1. It means that the antecedent and consequent are

TABLE 4: Runtime of the two algorithms on soybean dataset.

Support	0.75	0.76	0.77	0.78	0.79	0.80
Apriori algorithm	0.29	0.20	0.17	0.12	0.10	0.07
Improved algorithm	0.21	0.135	0.082	0.075	0.065	0.05

- (1) Gill-attachment = free \rightarrow veil-type = partial *support* = 0.9732382, *confidence* = 1, *lift* = 1
- (2) Veil-color = white \rightarrow veil-type = partial *support* = 0.9745126, *confidence* = 1, *lift* = 1
- (3) Ring-number = one \rightarrow veil-type = partial *support* = 0.9336052, *confidence* = 1, *lift* = 1
- (4) Gill-attachment = free, Veil-color = white \rightarrow veil-type = partial *support* = 0.9722187, *confidence* = 1, *lift* = 1
- (5) Gill-attachment = free, Ring-number = one \rightarrow veil-type = partial *support* = 0.9091372, *confidence* = 1, *lift* = 1
- (6) Veil-color = white, Ring-number = one \rightarrow veil-type = partial *support* = 0.9081178, *confidence* = 1, *lift* = 1
- (7) Gill-attachment = free, Veil-color = white, Ring-number = one \rightarrow veil-type = partial *support* = 0.9081178, *confidence* = 1, *lift* = 1

ALGORITHM 2: A subset of the mining result of mushroom dataset.

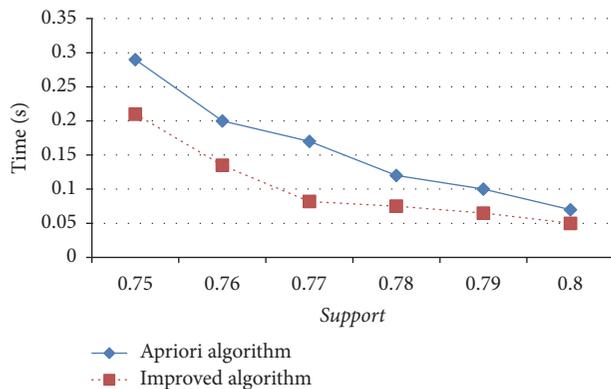


FIGURE 2: Runtime of the two algorithms on soybean dataset.

independent, and these association rules are not the rules this paper expect even though they have high *support* and *confidence*.

5. Conclusion

To avoid the weakness of apriori algorithm, this paper proposes an improved algorithm based on matrix and applies the improved algorithm to agricultural datasets. Experimental results show that the improved algorithm can efficiently discover useful association rules for the reason that database will be scanned only one time and that the data to deal with is getting smaller and smaller with the algorithm running. The improved algorithm is more applicable when the database is huge. But it is not that efficient compared with apriori when the database is not that large due to the fact that the scale data to deal with is small but the improved algorithm has an extra operation to covert the database to a matrix. Further research should be focused on the optimization of the proposed algorithm so as to further improve the efficiency

when applied to big data. Algorithm parallelization can be taken into account. Therefore our future work is to improve our algorithm so as to be applicable for more kinds of database. Besides, new evaluation criteria can be used to optimize our mining result.

Conflicts of Interest

There are no conflicts of interest.

References

- [1] Y. Pang, Y. Jin, Y. Zhang, and T. Zhu, "Collaborative filtering recommendation for MOOC application," *Computer Applications in Engineering Education*, vol. 25, no. 1, pp. 120–128, 2017.
- [2] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.
- [3] L. Qi, W. Dou, and X. Zhang, "Service recommendation based on social balance theory and collaborative filtering," in *Proceedings of the Intelligence and Lecture Notes in Bioinformatics, 14th International Conference*, vol. 9936 of *Lecture Notes in Computer Science*, pp. 637–645, Springer International Publishing, Basel, Switzerland, 2016.
- [4] B. Kyoungsoo and S.-G. Cheongju, "Social group recommendation based on dynamic profiles and collaborative filtering," *Neurocomputing*, vol. 209, pp. 3–13, 2016.
- [5] Y. Gao, "Collaborative filtering recommendation model based on normalization method," *International Journal of Grid and Distributed Computing*, vol. 9, no. 10, pp. 291–300, 2016.
- [6] M. Sun, F. Li, J. Lee et al., "Learning multiple-question decision trees for cold-start recommendation," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pp. 445–454, ACM, Rome, Italy, February 2013.
- [7] T.-H. Ma, L.-M. Guo, M. Li, M.-L. Tang, Y. Tian, and A. Mznah, "A collaborative filtering recommendation algorithm based on hierarchical structure and time awareness," *IEICE Transactions on Information and Systems*, no. 6, pp. 1512–1520, 2016.

- [8] H. L. dos Santos, C. Cechinel, R. M. Araujo, and M.-Á. Sicilia, "Clustering learning objects for improving their recommendation via collaborative filtering algorithms," *Communications in Computer and Information Science*, vol. 544, pp. 183–194, 2015.
- [9] P. Krupa, A. Thakkar, C. Shah, and K. Makvana, "A state of art survey on shilling attack in collaborative filtering based recommendation system," *Smart Innovation, Systems and Technologies*, vol. 50, pp. 377–385, 2016.
- [10] P. Mirko and A. Fabio, "Kernel based collaborative filtering for very large scale top-N item recommendation," in *Proceedings of the 24th European Symposium on Artificial Neural Networks*, pp. 11–16, 2016.
- [11] Y. Shen, T.-G. Lv, X. Chen, and Y.-D. Wang, "A collaborative filtering based social recommender system for E-commerce," *International Journal of Simulation: Systems, Science and Technology*, vol. 17, no. 22, pp. 91–96, 2016.
- [12] S. Rossi, F. Barile, D. Improta, and L. Russo, "Towards a collaborative filtering framework for recommendation in museums: from preference elicitation to group's visits," in *Proceedings of the 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks, EUSPN 2016 / The 6th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare, ICTH '16*, pp. 431–436, Elsevier, London, UK, September 2016.
- [13] B. Urszula, "Differential evolution in a recommendation system based on collaborative filtering," in *Proceedings of The Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, Lecture Notes in Computer Science, pp. 113–122, Springer International Publishing.
- [14] W. Ebisa and W. Vitor, "User-based collaborative filtering recommender systems approach in industrial engineering curriculum design and review process," in *Proceedings of the ASEE Annual Conference and Exposition*, 2016.
- [15] J. Jinhyun, B. Sangwon, and P. Geunduk, "Implementation of a recommendation system using association rules and collaborative filtering," in *Proceedings of the Proceedings of the 4th International Conference on Information Technology and Quantitative Management, ITQM '16*, pp. 944–952, 2016.
- [16] Y. K. Ng, "Recommending books for children based on the collaborative and content-based filtering approaches," in *Proceedings of the Computational Science and Its Applications-ICCSA '16*, Lecture Notes in Computer Science, pp. 302–317, Springer International Publishing.
- [17] H. H. Qiu, Y. Liu, Z. J. Zhang, and G. X. Luo, "An improved collaborative filtering recommendation algorithm for microblog based on community detection," in *Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 876–879, IEEE, Kitakyushu, Japan, August 2014.
- [18] K. Kim and H. Ahn, "Recommender systems using cluster-indexing collaborative filtering and social data analytics," *International Journal of Production Research*, vol. 55, no. 17, pp. 5037–5049, 2017.
- [19] C.-Y. Li and K.-J. He, "An optimized map reduce for item-based collaborative filtering recommendation algorithm with empirical analysis," *Concurrency Computation*, 2017.
- [20] N. Polatidis and C. K. Georgiadis, "A dynamic multi-level collaborative filtering method for improved recommendations," *Computer Standards & Interfaces*, vol. 51, pp. 14–21, 2017.
- [21] R. L. Palak, "An effective collaborative filtering based method for movie recommendation," *Advances in Intelligent Systems and Computing*, vol. 506, pp. 149–159, 2017.
- [22] M. Liu, Z. Zeng, W. Pan, X. Peng, Z. Shan, and Z. Ming, "Hybrid One-Class Collaborative Filtering for Job Recommendation," in *Smart Computing and Communication*, Lecture Notes in Computer Science, pp. 267–276, Springer International Publishing, 2017.
- [23] M. Sridevi and R. R. Rao, "An enhanced personalized recommendation utilizing expert's opinion via collaborative filtering and clustering techniques," in *Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT)*, pp. 1–4, IEEE, Coimbatore, India, August 2016.
- [24] H. Zhang, I. Ganchev, N. S. Nikolov, and M. O'droma, "A trust-enriched approach for item-based collaborative filtering recommendations," in *Proceedings of the 12th IEEE International Conference on Intelligent Computer Communication and Processing, ICCP '16*, pp. 65–68, September 2016.
- [25] L.-Y. Dong, G.-L. Zhu, Q. Zhu, and Y.-L. Li, "Research on collaborative filtering recommendation based on k-means clustering," *ICIC Express Letters*, pp. 2493–2498, 2016.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

