

PAPER • OPEN ACCESS

Application of Latent Factor Model on a Restaurant Menu Recommendation System

To cite this article: Noor Ifada *et al* 2020 *J. Phys.: Conf. Ser.* **1569** 022062

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Application of Latent Factor Model on a Restaurant Menu Recommendation System

Noor Ifada ^{1,*}, Zemmy Muamal Humaidi ², Mochammad Kautsar Sophan ³

^{1,2,3} Informatics Department, University of Trunojoyo Madura, Indonesia, 69162

* Corresponding author: noor.ifada@trunojoyo.ac.id

Abstract. A restaurant menu recommendation system is becoming a necessity following the trend of eating out at the restaurants. The task of such a system is to generate a top- N list of menus that may be of interest to a customer, in which the customer previous rating behavior is used as the learning model. In this paper, we apply the SVD (Singular Vector Decomposition) latent factor model as the learning technique of the recommendation system. Beforehand, we implement the mean imputation technique to fill in the missing rating entries so that SVD can also deal with the new customer that has no rating record in the system. Evaluation on a real-world restaurant menu recommendation dataset shows that our recommendation system is able to generate a top-10 list of menu recommendations to a target customer and that the results of the low-rank approximation using SVD are comparable with that of the full-rank.

Keywords: latent factor model; low-rank approximation; recommendation system; restaurant menu; SVD; Singular Vector Decomposition

1. Introduction

Nowadays, eating out at restaurants has become a lifestyle [1, 2]. A restaurant is a business that sells various menus of food and drinks to its customers. The bigger the business, the more menus are offered. Consequently, choosing which menus to be ordered can be challenging for the customers since the choice of the menus provided is overwhelming.

Recommendation systems help users to tackle the problem of having to find items suit their preference by generating a set of a personalized list of recommendations that might be of interest its users by learning through their previous rating activities [3, 4]. In this case, the task of a restaurant menu recommendation system is to generate a list of menus that may be of interest to a customer by learning from his previous rating behavior.

Collaborative Filtering (CF) is a very popular learning approach in RS [3-6], in which the latent factor models are considered to perform the best [4]. Such models learn from the rating data to derive the latent relationships between its dimensions. One of the well-known methods of the latent factor model is the Singular Value Decomposition (SVD) [7-9] in which the low-rank approximation technique is implemented. Low-rank approximation means that to generate the list of recommendation, we use the reduced size of the dimension of the latent factors instead of their full size. On the other hand, CF is known to typically suffer from a sparsity issue that impacts the recommendation performance [5, 10-14]. The problem occurs since the unobserved rating entries are commonly dominant in numbers compared to the observed data.



Our work is focusing on a restaurant menu recommendation system that applies the low-rank approximation using SVD as the latent factor model to learn from the rating data. To deal with the sparsity issue, we implement the mean imputation technique [15] that is to replace the non-observed rating entries with the mean rating of each menu, resulting in dense rating data.

Experiment results in a real-world restaurant menu rating dataset show that the recommendation system can generate a top- N list of menu recommendations to a target customer. We also show that the low-rank approximation using SVD works effectively as its results are comparable with that of the full-rank.

The summary of our contributions is as follows: (1) a restaurant menu recommendation system that implements low-rank approximation using SVD as the latent factor model to learn from the menu rating data, (2) and the implementation of mean imputation technique to deal with the missing rating entries.

2. Literature Review

The task of a recommendation system is to generate a top- N list of recommendations to the users. A widely approach used in the recommendation systems is Collaborative Filtering (CF). It can be implemented based on the memory-based or model-based approach [4]. The memory-based is a CF approach that employs the users or items similarities to generate the list of recommendations to a target user [4]. On the other hand, the model-based is a CF approach that develops a model to learn a pattern based on the training data and then uses it to generate the list of recommendations [4]. Several approaches can be implemented to create the model using Bayesian [16], clustering [17-19], and latent factor model [7-9, 20, 21] techniques. The later performs the best in compared to others[4] that we apply the technique to our restaurant menu recommendation system.

The latent factor model can be solved using Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Probabilistic Matrix Factorization (PMF), and Non-Negative Matrix Factorization (NMF) methods [4]. Out of those four methods, the SVD is the most popular one [9]. For this reason, our study applies the low-rank approximation using SVD latent factor model as the learning technique of the recommendation system.

The CF approach commonly suffers from the sparsity problem, which impacts the recommendation quality [10]. This issue can be tackled by using imputation [11, 15, 22] or clustering [11-14, 23, 24] technique. The former is more straightforward, and thus it is simpler compared to the later. Therefore, our study implements the mean imputation technique [15] to deal with the sparsity issue in the recommendation model.

3. Applying Latent Factor Model on a Restaurant Menu Recommendation System

3.1. Overview

Our work focuses on building a restaurant menu recommendation system that applies a latent factor model. To address the sparsity issue of customer-menu rating data, we use the mean rating of each menu to impute the missing values rating matrix before applying a latent factor model technique. Low-rank latent factor matrices of customer and menu are then computed by reducing the dimension of the latent factors. Figure 1 shows the framework of our study.

3.2. Preliminary

The rating data consists of observed entries which form the binary correlations between customers and menus. Each rating score represents the customer's level of preference towards menus. Let $U = \{u_1, u_2, u_3, \dots, u_m\}$ be the set of m customers and $I = \{i_1, i_2, i_3, \dots, i_n\}$ be the set of n menus. The binary correlation of rating data is modeled as a rating matrix of $R \in \mathbb{R}^{m \times n}$, in which r_{ui} denotes the rating given by customer u to menu i . Whereas U_i indicates the set of customers who have rated menu i . Figure 2(a) presents the toy examples of a rating matrix $R \in \mathbb{R}^{3 \times 4}$ where $U = \{u_1, u_2, u_3\}$ and $I = \{i_1, i_2, i_3, i_4\}$. Therefore, $U_1 = \{1,2\}$, $U_2 = \{1\}$, $U_3 = \{2\}$, and $U_4 = \{1, 3\}$.

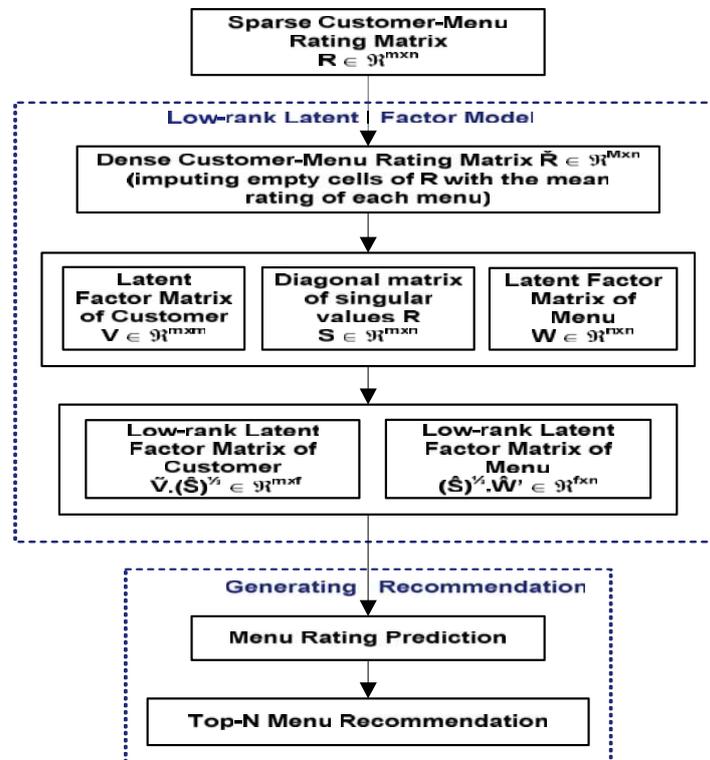


Figure 1. Framework of the application of low-rank approximation using a latent factor model on a restaurant menu recommendation system

		Menu			
		i_1	i_2	i_3	i_4
Customer	u_1	1	5	0	2
	u_2	4	0	3	0
	u_3	0	0	0	4

(a)

		Menu			
		i_1	i_2	i_3	i_4
Customer	u_1	1	5	3	2
	u_2	4	5	3	3
	u_3	2.5	5	3	4

(b)

Figure 2. Examples of: (a) Sparse customer-menu rating matrix R and (b) Dense customer-menu rating matrix \hat{R}

3.3. Low-rank Approximation using Singular Value Decomposition (SVD)

In this paper, we implement the Singular Value Decomposition (SVD) as the latent factor model due to its popularity [9]. SVD factors a matrix $R \in \mathbb{R}^{m \times n}$ into three matrices as [4]:

$$R = V \cdot D \cdot W' \quad (1)$$

where $V \in \mathbb{R}^{m \times m}$ and $W \in \mathbb{R}^{n \times n}$ are the orthogonal matrices that respectively represent the latent factor of customer and menu, and $S \in \mathbb{R}^{m \times n}$ is the diagonal matrix of singular values of R which satisfy $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{m,n\}} \geq 0$. The SVD of a matrix is visualized in Figure 3.

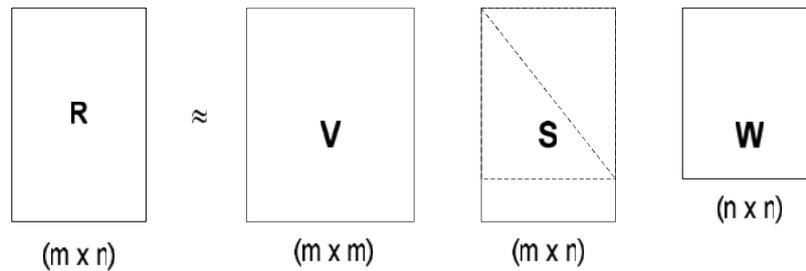


Figure 3. Visualization of SVD technique on a matrix

To deal with the sparse customer-menu rating data, we impute the missing value in R with the mean rating of each menu [15] to result in \hat{R} :

$$\hat{r}_{ui} := \begin{cases} r_{ui} & \text{if } r_{ui} \neq 0 \\ \frac{\sum_{k \in U_i} r_{uk}}{|U_i|} & \text{otherwise} \end{cases} \quad (2)$$

By this, the SVD technique is implemented on \hat{R} , instead of R . Figure 2(b) shows the sparse R (Figure 2) now becomes the dense \hat{R} .

Algorithm: Low-rank Approximation using SVD Technique

Input: Training set $D_{train} := U \times I$, the size of rank f

Output: Low-rank latent factor matrices $\hat{V}\sqrt{\hat{S}}$ and $\sqrt{\hat{S}}\hat{W}$

Process:

1. Construct customer-menu matrix $R \in \mathbb{R}^{m \times n}$ using rating data of D_{train} where $m = |U|$ and $n = |I|$
2. Calculate the mean rating of each menu. Fill in the empty cell of R with the value of mean rating accordingly, resulting dense customer-menu matrix $\hat{R} \in \mathbb{R}^{m \times n}$
3. Apply SVD technique to matrix \hat{R} to produce V, S, W'
4. Reduce the matrix S to dimension f
5. Compute the square root of the reduced matrix \hat{S} to obtain $\sqrt{\hat{S}}$
6. Compute the two low-rank latent factor matrices: $\hat{V}\sqrt{\hat{S}}$ of size $m \times f$ and $\sqrt{\hat{S}}\hat{W}'$ of size $f \times n$

Figure 4. Algorithm of low-rank approximation using SVD Technique

Matrix R can be approximated using the low-rank SVD within three steps [7] by setting the rank $f \ll \min\{m, n\}$. First, the diagonal matrix S is reduced to dimension f . Second, compute the square root of the reduced diagonal matrix \hat{S} to obtain $\sqrt{\hat{S}}$. Last, compute the two low-rank latent factor matrices of customer and menu, i.e., $\hat{V}\sqrt{\hat{S}} \in \mathbb{R}^{m \times f}$ and $\sqrt{\hat{S}}\hat{W}' \in \mathbb{R}^{f \times n}$. Figure 4 presents the complete algorithm of the low-rank approximation using the SVD technique.

3.4. Generating Menu Recommendation

The menu rating prediction is calculated by using the low-rank latent factor matrices of customer and menu generated in the previous section. In this case, the rating prediction of a target customer u to menu i is computed by the dot product of the u^{th} row of $\hat{V} \cdot \sqrt{\hat{S}}$ and i^{th} column of $\sqrt{\hat{S}} \cdot \hat{W}'$:

$$p_{ui} = \sum_{k=1}^f \left(\hat{V} \cdot \sqrt{\hat{S}} \right)_{uk} \cdot \left(\sqrt{\hat{S}} \cdot \hat{W}' \right)_{ki} \quad (3)$$

Afterwards, the menus with high predicted ratings are recommended as a top- N list of menu recommendations $Top(N)$ to each target customer u .

4. Empirical Analysis

4.1. Dataset and Experiment Design

For the experiments, we manually collected the customer-menu rating data from the *Laras Liris Resto & Coffee Shop*¹ from January 2018 until September 2018. The collection consists of 100 customers, 119 menus, and 1344 menu rating.

To evaluate the recommendation quality, we implement the 4-fold cross-validation where each fold is randomly split into a 75% training set D_{train} and a 25% test set D_{test} . The recommendation quality is reported over the average results of all folds.

4.2. Evaluation Metric

We measure the recommendation quality of each target customer u based on the following F1-Score formulation:

$$F1 - Score(N) = \frac{2 \cdot (Precision(N) \cdot Recall(N))}{Precision(N) + Recall(N)} \quad (4)$$

where

$$Precision(N) = 100 \cdot \frac{|Top(N) \cap GT|}{N} \quad (5)$$

$$Recall(N) = 100 \cdot \frac{|Top(N) \cap GT|}{|GT|} \quad (6)$$

where $Top(N)$ is the top- N list of restaurant menu recommendations and GT is the list of ground-truth menus in D_{test} . The reported F1-Scores are the average scores of all customers in the D_{test} .

4.3. Experiment Result

The restaurant application is built using the *Laravel PHP Framework*², and the SVD technique is implemented using the *Real Matrices Tools Class Package V 1.0.0*³. The recommendation process is conducted whenever a customer opens the restaurant application.

Figure 5(a) and (b) respectively show the examples of the back-end results of menu rating prediction and top-10 menu recommendation for a target customer u_{119} . Meanwhile, Figure 5(c) shows the front-end results, i.e., web page display, of the top-10 menu recommendation. Note that the list of menu rating prediction in Figure 5 is cropped for convenient presentation. Furthermore, since the total number of customers recorded of the menu rating data is 100, u_{119} indicates that the customer is new in the system and that he will be recorded as the 120th customer in the database – given that the indexing of PHP starts from 0. The generation of recommendations for customers that have no record in the D_{train} is possible since we implement the imputation technique formulated in Equation (2).

To evaluate the sensitivity of low-rank dimension f , we conduct experiments of various f by using the size of $H \in \min\{m, n\}$ multiply by a total grid of coefficients between 0.1 and 1 with an interval

¹ <https://www.facebook.com/pages/Cafe-laras-liris-Lamongan/168047237302614>

² <https://laravel.com/>

³ <https://www.phpclasses.org/blog/package/10348/post/1-PHP-Matrix-Math-Library.html>

of 0.1, resulting $f \in \{(0.1 * H), (0.2 * H), (0.3 * H), (0.4 * H), (0.5 * H), (0.6 * H), (0.7 * H), (0.8 * H), (0.9 * H), (1.0 * H)\}$. Note that $f = 1.0 * H$ means that the latent factor matrices are not truncated. Figure 6 shows that the best recommendation performance is achieved when $f \in \{(0.8 * H), (0.9 * H), (1.0 * H)\}$. These results confirm that the low-rank approximation technique (i.e., $f \in \{(0.8 * H), (0.9 * H)\}$) can generate a comparable recommendation quality as the full-rank SVD (i.e., $f = 1.0 * H$).

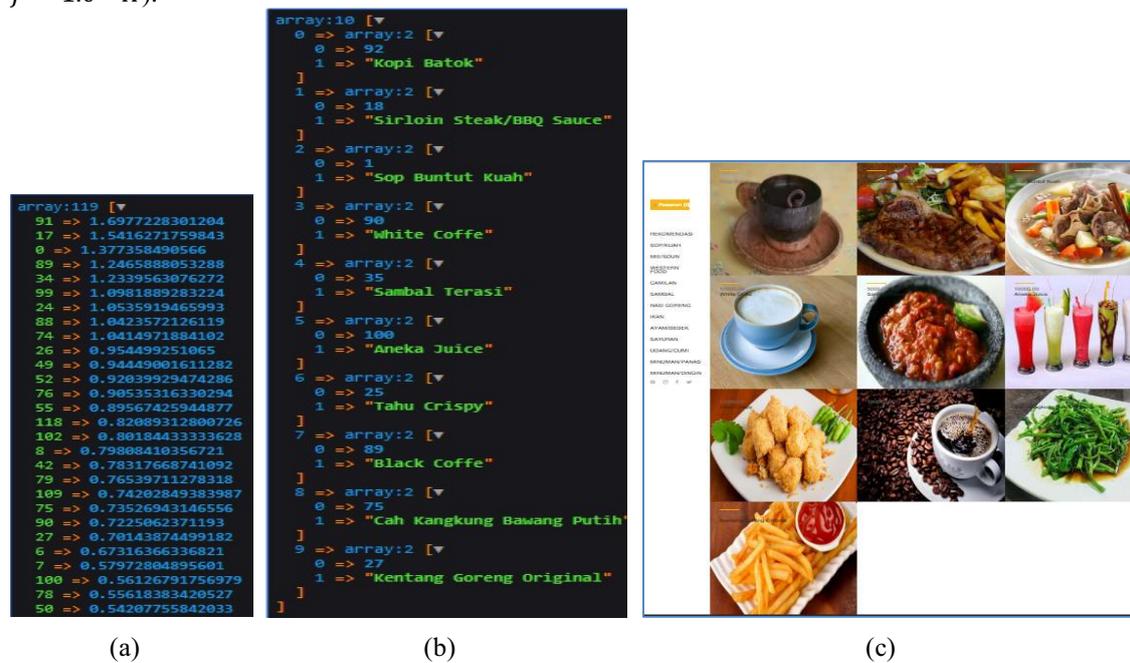


Figure 5. Example of a target customer u_{119} : (a) back-end results of the menu rating prediction, (b) back-end results of the top-10 menu recommendation, and (c) front-end display of the top-10 menu recommendation

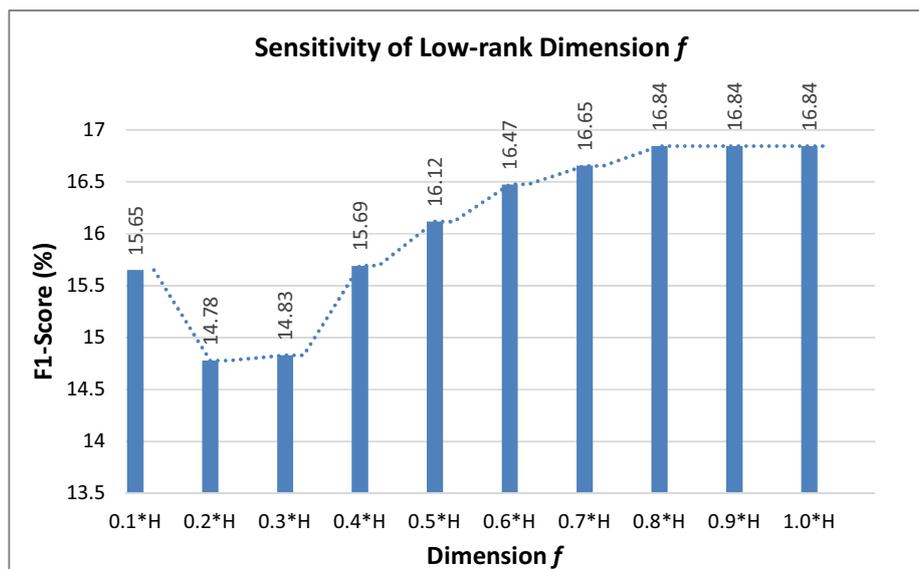


Figure 6. Low-rank dimension f sensitivity analysis

5. Conclusion

Our study applies the low-rank approximation using SVD as the latent factor learning model on restaurant menu recommendation system. To deal with the sparsity problem, we implement the mean imputation technique to fill in the missing rating entries to generate the dense data. The empirical analysis shows that the restaurant menu recommendation system is able to generate a top-10 list of menu recommendations to a target customer. Moreover, we also show that the results of the low-rank approximation using SVD are comparable with that of the full-rank.

6. References

- [1] Suryana A, Ariani M, and Lokollo E M 2017 The role of modern markets in influencing lifestyles in indonesia *Jurnal Penelitian dan Pengembangan Pertanian* **27** 10-15
- [2] Ali J and Nath T 2013 Factors affecting consumers' eating-out choices in india: Implications for the restaurant industry *Journal of Foodservice Business Research* **16** 197-209
- [3] Konstan J A and Riedl J 2012 Recommender systems: From algorithms to user experience *User Modeling and User-Adapted Interaction* **22** 101-23
- [4] Aggarwal C C 2016 *Recommender systems: The textbook* (Switzerland: Springer International Publishing)
- [5] Sarwar B, Karypis G, Konstan J, and Riedl J 2001 Item-based collaborative filtering recommendation algorithms *Proc. The 10th International Conference on World Wide Web (Hong Kong)* pp 285-95
- [6] Gong S J 2009 Employing user attribute and item attribute to enhance the collaborative filtering recommendation *Journal of Software* **4** 883-90
- [7] Sarwar B, Karypis G, Konstan J, and Riedl J 2000 Application of dimensionality reduction in recommender system - a case study (Minneapolis: Minnesota University Minneapolis Department of Computer Sciercer)
- [8] Koren Y, Bell R, and Volinsky C 2009 Matrix factorization techniques for recommender systems *Computer* **42** 30-37
- [9] Bokde D, Girase S, and Mukhopadhyay D 2015 Matrix factorization model in collaborative filtering algorithms: A survey *Procedia Computer Science* **49** 136-46
- [10] Papagelis M and Plexousakis D 2005 Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents *Engineering Applications of Artificial Intelligence* **18** 781-89
- [11] Sarwar B M, Karypis G, Konstan J, and Riedl J 2002 Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering *Proc. The 5th International Conference on Computer and Information Technology (Dhaka, Bangladesh)* pp 23-28
- [12] Koochi H and Kiani K 2016 User based collaborative filtering using fuzzy c-means *Measurement* **91** 134-39
- [13] Kumar N P and Fan Z 2015 Hybrid user-item based collaborative filtering *Procedia Computer Science* **60** 1453-61
- [14] Phorasim P and Yu L 2017 Movies recommendation system using collaborative filtering and k-means *International Journal of Advanced Computer Research* **7** 52-59
- [15] Han J, Kamber M, and Pei J 2012 *Data mining: Concepts and techniques* (Waltham, USA: Morgan Kaufmann)
- [16] Alper M E and Ögüdücü S G 2012 Personalized recommendation in folksonomies using a joint probabilistic model of users, resources and tags *Proc. The 11th International Conference on Machine Learning and Applications (ICMLA) (Boca Raton, Florida)* pp 368-73
- [17] Begelman G, Keller P, and Smadja F 2006 Automated tag clustering: Improving search and exploration in the tag space *Proc. Collaborative Web Tagging Workshop at WWW2006 (Edinburgh, Scotland)* pp 15-33
- [18] Pan R, Dolog P, and Xu G 2012 Knn-based clustering for improving social recommender systems *Proc. International Workshop on Agents and Data Mining Interaction (Valencia, Spain)* pp 115-25

- [19] Shepitsen A, Gemmell J, Mobasher B, and Burke R 2008 Personalized recommendation in social tagging systems using hierarchical clustering *Proc. The 2008 ACM Conference on Recommender Systems* pp 259-66
- [20] Koren Y 2008 Factorization meets the neighborhood: A multifaceted collaborative filtering model *Proc. The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Las Vegas, Nevada, USA)* pp 426-34
- [21] Koren Y and Sill J 2011 Ordrec: An ordinal model for predicting personalized item rating distributions *Proc. The 5th ACM conference on Recommender Systems (Chicago, Illinois, USA)* pp 117-24
- [22] Rendle S, Freudenthaler C, Gantner Z, and Schmidt-Thieme L 2009 Bpr: Bayesian personalized ranking from implicit feedback *Proc. The 25th Conference on Uncertainty in Artificial Intelligence (Montreal, Quebec, Canada)* pp 452-61
- [23] Dakhel G M and Mahdavi M 2011 A new collaborative filtering algorithm using k-means clustering and neighbors' voting *Proc. The 11th International Conference on Hybrid Intelligent Systems* pp 179-84
- [24] Wei S, Ye N, Zhang S, Huang X, and Zhu J 2012 Collaborative filtering recommendation algorithm based on item clustering and global similarity *Proc. The 5th International Conference on Business Intelligence and Financial Engineering (BIFE) (Lanzhou, China)* pp 69-72