



# Framework of dynamic recommendation system for e-shopping

Shihab Uddin Tareq<sup>1</sup> · Md. Habibullah Noor<sup>1</sup> · Chinmay Bepery<sup>2</sup>

Received: 20 February 2018 / Accepted: 7 November 2019 / Published online: 18 November 2019  
© Bharati Vidyapeeth's Institute of Computer Applications and Management 2019

**Abstract** The popularity of online shopping is growing rapidly in modern virtual market. Generally, customers take decision to purchase goods based on their basic need and relative need. Shopkeepers play an important role to influence the customers in real market. Recommendation engine is nothing but a good automated shopkeeper. In this paper, we propose a model of dynamic recommendation system (DRS) for online market. Our proposed technique provides an intelligent solution model to overcome the problems of customers' rating and their feedback by integrating market basket analysis, frequent item mining, bestselling items and customer personalization.

**Keywords** Recommendation · Collaborative filtering · Content based filtering · Data sparsity · Scalability · Demographic · Robustness · Serendipity

## 1 Introduction

People are familiar with recommendation system from ancient age. Computerized recommendation system was first introduced by “Jussi karlgren” in a technical report [1, 2]. Generally, people depend on recommendations delivered by another people through their words and activities [3, 4]. In real market, when customers ask for anything seller shows not only the demanded product but also shows some other similar products that customer may want. But in virtual market, there is opportunity to do this. Now-a-days in the context of online shopping, few online shops use attaining some sort of recommendation. If we create and implement it properly, it may boost up income significantly. We know that when we buy a product from online shop, they suggest us for rating their product. As this is not a mandatory field, majority skip this option. From literature, existing recommendation system recommends a product based on user rating. But there are some problems in the user rating-based recommendation system like data sparsity [5, 6], scalability [7, 8], shilling attack [9, 10].

### 1.1 Data sparsity

The sparsity come in a variety of state, especially the cold start problem occurs when new user or item has just come in the device, it is tough to give recommendation ones because there is not enough information. New items shouldn't be suggested until some users rate it [5, 6].

### 1.2 Scalability

When the user and item increase tremendously in the system then it is so difficult to make recommendation. Because in the collaborative filtering sustain serious

---

✉ Md. Habibullah Noor  
habib.cse.pstu@gmail.com

Shihab Uddin Tareq  
tareq.jkt94@gmail.com

Chinmay Bepery  
chinmay.cse@gmail.com

<sup>1</sup> Department of Computer Science and Engineering,  
Patuakhali Science and Technology University, Patuakhali,  
Bangladesh

<sup>2</sup> Department of Computer Science and Information  
Technology, Faculty of Computer Science and Engineering,  
Patuakhali Science and Technology University, Patuakhali,  
Bangladesh

scalability problems, with computational tools going behind practical or sufficient levels [7, 8].

### 1.3 Shilling attacks

If people can provide tons of positive rating about their own products and negative rating for their competitors. Exactly where anybody can offer suggestions, individuals can offer a lot of good suggestions regarding personal supplies as well as damaging suggestions for his or her rivals [9, 10].

There are two important ways of recommendation system:

- Collaborative filtering.
- Content-based filtering.

### 1.4 Collaborative filtering

Collaborative filtering algorithm considers “User Behavior” just for recommending items. They exploit routine of other user’s ratings knowledge. Other user’s behavior and preferences throughout the items are used to recommend what to the new users [11, 12]. Collaborative filtering predicts the eye of products for an active customer while using aggregated rating information of the likeminded customers within the historical database [13]. All the outcomes will be opinion explicating as to the location customer would like or dislike exact precise supplement and also a good recommended component [14].

### 1.5 Content-based filtering

content-based systems referred to as cognitive filtering. Content based filtering depends upon the uses choices [15]. The feature of items can be mapped with feature of users for you to obtain user-item likeness. The top matched pairs are listed as recommendations. From these matrix, we can determine the recommendation to users for individual product according to their choices [16, 17].

## 2 Dynamic recommendation system

A successful model has several parameters such as user demographics, robustness and serendipity [18]. Recommendation can be provided towards user by their demographics data [19]. Robustness indicates the issue of hype that must be identified in recommendation system [20]. The serendipity is a measurement of “how amazing the recommendation are” [21]. By satisfying these parameters, an intelligent recommendation model is proposed in this paper regarding the following issues (Fig. 1):

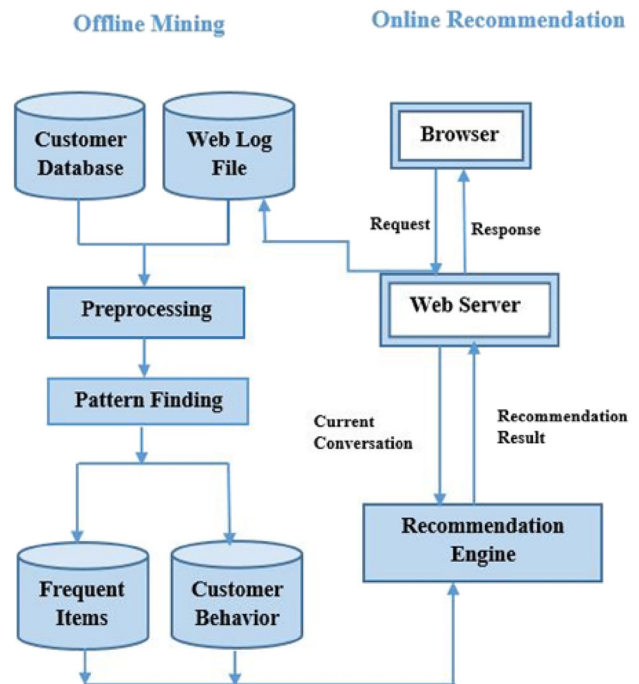


Fig. 1 System architecture for user recommendation

- Avoid user rating.
- Consider frequent items.
- Consider the customer searching/browsing history.
- Consider best-selling item.
- Consider data sparsity issue.
- Consider scalability issue.

According to our model recommendation system architecture is divided into two main parts i.e:

- Offline mining.
- Online recommendation.

In offline mining, the system manipulates the customer database and web log file. After analyzing customer data, it generates association rule and find frequent item. On the other hand, web log file contains all the information according to the user activity within the site [22]. Here, customer database is personalized and updated according to the user account. And the web log file is created and updated by user account and also individual device account. In case of web log file, user account supersedes the device account. After analyzing these data, the customer behavior (i.e. what one likes, what one may want etc.) will be traced out.

All results from offline mining are stored in recommendation engine. In online recommendation, when a user hit anything to the browser then the browser send a request to the server. The server sends the information to the recommendation engine and recommendation engine analyze the current status of the users and produce the immediate response for the users [23] (Fig. 2).

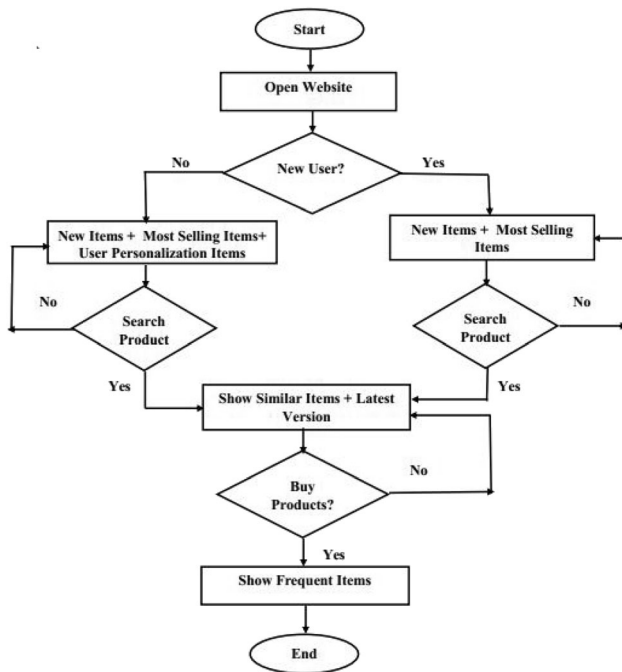


Fig. 2 Model flow chart

Working principles of proposed dynamic recommendation system is given as:

**User interact with web interface**

**(1) If a new customer**

Fetch most selling items and new items

**Else**

Fetch the items based on user personalization  
And fetch most selling items and new items

End if

**(2) If search an item**

Fetch specific item and also fetch Similar items based on the features of searching item and the Latest version of searching item.

**Else**

Go to step 1

End if

**(3) If buy products**

Show most frequent items of these product

**Else**

Go to step 1

End if

Web interface can be categories into two parts: interface for new user and interface for existing user. Initially, each new device works as a new customer until the user provides authentication. After authentication, personalization of customer is used for recommendation. Initially, the proposed model recommends the most selling items and the new items for new user. A set of new products are shown in front of new users. A new product is recognized based on their launching time. Also, a new item can be migrated into the set of best-selling items. A best-selling item is determined by the weight of the item. When the weight of the item is crossed the threshold value, the item is migrated to best-selling item. The threshold value will be changed based on the number of items in best-selling set. The weight of the item is calculated by the following heuristic function:

$$\begin{aligned}
 \text{weight}_i = & \text{count}_i(\text{hour}) \times 0.25 + \text{count}_i(\text{day}) \times 0.20 \\
 & + \text{count}_i(\text{week}) \times 0.15 + \text{count}_i(\text{month}) \times 0.10 \\
 & + \text{count}_i(3\_month) \times 0.08 + \text{count}_i(6\_month) \times 0.07 \\
 & + \text{count}_i(9\_month) \times 0.06 + \text{count}_i(12\_month) \times 0.05 \\
 & + \text{count}_i(\text{life\_time}) \times 0.04,
 \end{aligned}
 \tag{1}$$

where,  $i$  is item id and  $i = \{1,2,3,\dots,n\}$ ,  $\text{count}_i(\text{hour})$  Total number sells in the last hour for product  $i$ ,  $\text{count}_i(\text{day}) = \text{absolute\_count}_i(\text{day}) - \text{count}_i(\text{hour})$ ,  $\text{absolute\_count}_i(\text{day}) =$  Total number sells in the last day for product  $i$ , Similarly,  $\text{count}_i(\text{month}) = \text{absolute\_count}_i(\text{month}) - \text{count}_i(\text{week}) - \text{count}_i(\text{day}) - \text{count}_i(\text{hour})$ .

The weight of item  $i$  is distributed based on selling period from most recent to its initiation.

When the existing user interact with the web interface, the most selling items, new items and user personalization items are shown. To provide the personalized recommendation, we consider the parameters such as: user’s age, living area, gender, previous buying and browsing history.

In the case of weight calculation to identify the most selling item, both ‘Add to cart’ and ‘Payment’ categories are considered:

$$\text{“Addtocart”} + \text{“Payment”} = \text{“Checkout”}.
 \tag{2}$$

So that we take a set of top weighted products (ten products) that satisfy this equation.

Regarding the proposed model, if user search anything like a cellphone, but not a specific model then it shows products according to the most click items that is viewed by users at a huge time. If user search specific item, the system recommends similar items. User can look for an item by using any feature of that item. The system provides the targeted items according to the user personalization and weight of the item. Suppose, a user choose “Nokia 3” which price is 13,500 takas, so the feature of “Nokia 3”

will be displayed. But in the recommendation box, the system shows the similar smart phone like Samsung, Huawei, Lava etc. which configuration or outlook might be similar or better than “Nokia 3” and which price is closely related to 13,500 takas. The system provides the targeted items according to latest version of that product. Suppose, a user search “Samsung J5”. In the recommendation box we want to show him/her “Samsung J7” that is the latest version of Samsung J5”. So that there is a close possibility to buy a smartphone. If a customer buys any product, the system endorses the customer to buy the most frequent items. Suppose, a user buys a laptop, the recommendation engine provides most frequent items like key board, mouse, modem etc.

### 3 Discussion

The special features of our model are user personalization, frequent item mining, surprise element and avoid user rating. We have combined a bunch of technique rather than considering one to make better recommendations. For user personalization we analyze user demographic data from web log file. This model exercise FP growth algorithm to find out most frequent item. And for surprise element, the most prominent distinction of this model, we consider most selling item and new item by calculating their weight. For user personalization we analyze user demographic data from web log file. This model exercise FP growth algorithm to find out most frequent item. And for surprise element, the most prominent distinction of this model, we consider most selling item and new item by calculating their weight. To find most frequent items there are two algorithms. One is Apriori algorithm and another is FP-Growth algorithm. We used FP-Growth method. Because Apriori algorithm needs large time and memory so that it is optimal [24]. To generate association rule using frequent item mining we use “Rapid Miner Studio” (RM). We have used ‘Breakfast at the Frat’ dataset from [25, 26].

In Fig. 3 shows that how we generate association rule in rapid miner. According to Table 1 we see that for first row the support is 0.419 that mean in the total transaction, 41.9% possibility to buy (SPEND, UNITS, HHS) products together. Confidence 0.837 indicates if a customer buy “SPEND” product there is 83.7% possibility to buy “UNITS” and “HHS” products.

In Figs. 4 and 5 shows output association rule and its output graph that is generated by rapid miner. In current system there are some recommendation algorithms, but most of the algorithm cannot solve these three main problems (data sparsity, scalability, shilling attack) because they consider user rating for recommendation. Our proposed model can solve all these major problems of current

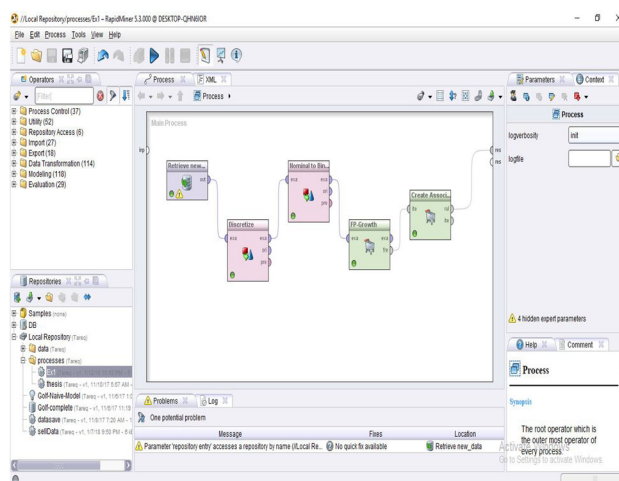


Fig. 3 Raw data input process

Table 1 Output support and confidence (rapid miner)

No.	Premises	Conclusion	Support	Confidence
1	SPEND	UNITS, HHS	0.419	0.837
2	SPEND	VISITS, UNITS, HHS	0.419	0.837
3	SPEND	VISITS, UNITS	0.424	0.848
4	SPEND	HHS	0.425	0.851
5	SPEND	VISITS, HHS	0.426	0.851
6	SPEND	UNITS	0.430	0.859
7	VISITS	SPEND, UNITS, HHS	0.419	0.850
8	SPEND	VISITS	0.432	0.852
9	UNITS	SPEND, HHS	0.419	0.854
10	UNITS	SPEND, VISITS, HHS	0.419	0.854
12	HHS	SPEND, UNITS	0.419	0.870
–	–	–	–	–
–	–	–	–	–
–	–	–	–	–
549	UNITS	SPEND, VISITS	0.424	0.875
550	VISITS, HHS	SPEND, UNITS	0.419	0.870

recommendation system. Table 2 shows the comparison between current algorithms and our algorithm.

### 4 Conclusion and future scope

Automatic recommendation system is very important for e-commerce site. Amazon says 40% of their sales increased using recommendation engine. A perfect dynamic recommendation engine then it would be helpful to increase the revenue of ecommerce business. In this paper we have proposed a DRS model which avoid the user rating system and overcome data sparsity, scalability, shilling attack



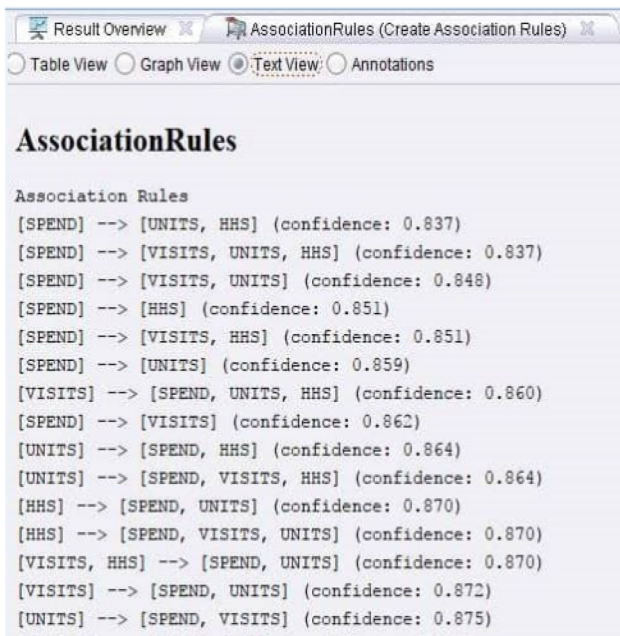


Fig. 4 Output association rule

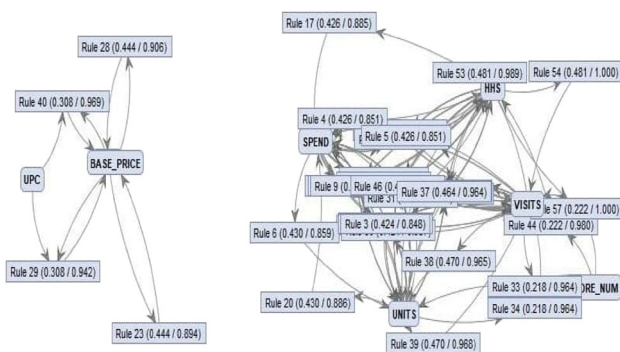


Fig. 5 Output association rule graph

Table 2 Comparison table

References	Solution of data sparsity problem	Solution of scalability problem	Solution of shilling attack problem
[22]	No	No	No
[23]	No	Yes	No
[26]	No	No	No
[27]	Yes	Yes	No
Our algorithm	Yes	Yes	Yes

problems. But we don't prepare a complete recommendation engine. Our main goal is to implement the model into a real e-commerce or online retailer.

## References

- Karlgren J (1990) An algebra for recommendations. Syslab Working Paper 179
- Karlgren J (1994) Newsgroup clustering based on user behavior-a recommendation algebra. SICS Research Report
- Resnick P, Varian HR (1997) Recommender systems. Commun ACM 40(3):56–58
- Hill W, Stead L, Rosenstein M, Furnas G (1995) Recommending and evaluating choices in a virtual community of use. In: Proceedings CHI '95 Proceedings of the SIGCHI conference on Human factors in computing systems, Denver, Colorado, USA, 7–11 May 1995. ACM Press/Addison-Wesley Publishing Co., pp 194–201
- Mulpuru S, Johnson C, Roberge D (2013) US online retail forecast, 2012–2017. Forrester Res
- Grčar M, et al (2005) Data sparsity issues in the collaborative filtering framework. In: 7th International workshop on knowledge discovery on the web, WebKDD 2005, Chicago, IL, USA, 21 Aug 2005. Revised Papers
- Ye F, Zhang H (2016) A collaborative filtering recommendation based on users' interest and correlation of items. In: 2016 International conference on audio, language and image processing (ICALIP), Shanghai, China, 11–12 July 2016. IEEE
- Herlocker JL et al (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inform Syst (TOIS) 22(1):5–53
- Chirita PA, Nejd W, Zamfir C (2005) Preventing shilling attacks in online recommender systems. In: Conference CIKM '05 Conference on information and knowledge management, Bremen, Germany, 31 Oct–5 Nov 2005. ACM
- Gunes I et al (2014) Shilling attacks against recommender systems: a comprehensive survey. Artif Intell Rev 42(4):768
- Linden Greg, Smith Brent, York Jeremy (2003) Amazon. com recommendations: item-to-item collaborative filtering. IEEE Internet Comput 7(1):76–80
- Grčar M et al (2005) Data sparsity issues in the collaborative filtering framework. In: 7th International workshop on knowledge discovery on the web, WebKDD 2005, Chicago, IL, USA, 21 Aug 2005. Revised Papers
- Zhao ZD, Shang MS (2010) User-based collaborative-filtering recommendation algorithms on hadoop. In: Third international conference on knowledge discovery and data mining, WKDD 2010, Phuket, Thailand, 9–10 Jan 2010. IEEE
- Resnick P, Neophytos I, Mitesh S, Peter B, John R (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina, USA, 22–26 Oct 1994. ACM, pp 175–186
- Pazzani MJ, Billsus D (2007) Content-based recommendation systems. The adaptive web. Springer, Berlin, pp 325–341
- Frias-Martinez E, Chen SY, Liu X (2009) Evaluation of a personalized digital library based on cognitive styles: adaptivity vs. adaptability. Int J Inform Manag 49:48–56
- Zhang H, et al (2016) A trust-enriched approach for item-based collaborative filtering recommendations. In: 2016 IEEE 12th international conference on intelligent computer communication and processing (ICCP), Cluj-Napoca, Romania, 8–10 Sept 2016. IEEE
- Sarwar B, et al (2000) Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM conference on Electronic commerce, Minneapolis, Minnesota, USA, 17–20 Oct 2000. ACM
- Ricci F, Rokach L, Shapira B (2011) Recommender systems handbook. Springer, Berlin

20. Konstan JA, Riedl J (2012) User modeling and user-adapted interaction. Springer, Berlin
21. Tan X, Yao M, Xu M (2006) An effective technique for personalization recommendation based on access sequential patterns. In: IEEE Asia-Pacific conference on services computing, APSCC'06, Guangzhou, China, 12–15 Dec 2006. IEEE
22. Zhao G, Luo S, He J (2009) Style matching model-based recommend system for online shopping. In: IEEE 10th international conference on computer-aided industrial design & conceptual design, CAID & CD 2009, Wenzhou, China, 26 Nov 2009. IEEE
23. Zhang F, Chang HY (2005) On a hybrid rule based recommender system. In: The Fifth international conference on computer and information technology, CIT 2005, Shanghai, China, 21–23 Sept 2005. IEEE
24. Woo J, Xu Y (2011) Market basket analysis algorithm with map/reduce of cloud computing. In: The 2011 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2011), Las Vegas, Nevada, USA, 18–21 July 2011
25. <https://www.dunnhumby.com/careers/engineering/sourcefiles>
26. Vishal B (2017) Collaborative filtering using data mining and analysis books.google.com. IGI Global, USA
27. Guo YY, Liu QC (2010) E-commerce personalized recommendation system based on multi-agent. In: 2010 Seventh international conference on fuzzy systems and knowledge discovery (FSKD), Yantai, Shandong, China, 10–12 Aug 2010, Vol 4. IEEE