Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

A Propound Hybrid Approach for Personalized Online Product Recommendations

Veer Sain Dixit, Shalini Gupta & Parul Jain

To cite this article: Veer Sain Dixit, Shalini Gupta & Parul Jain (2018) A Propound Hybrid Approach for Personalized Online Product Recommendations, Applied Artificial Intelligence, 32:9-10, 785-801, DOI: 10.1080/08839514.2018.1508773

To link to this article: https://doi.org/10.1080/08839514.2018.1508773

APPLIED ARTIFICIAL INTELLIGENCE
2018, VOL. 32, NOS. 9–10, 785–801
https://doi.org/10.1080/08839514.2018.1508773

Taylor & Francis
Taylor & Francis Group

A Propound Hybrid Approach for Personalized Online Product Recommendations

Veer Sain Dixit, Shalini Gupta, and Parul Jain

Department of Computer Science, ARSD College, University of Delhi, New Delhi, India

ABSTRACT

The main aim of e-commerce websites is to turn their visitors into customers. For this purpose, recommender system is used as a tool that helps in turning clicks into purchases. Obtaining explicit ratings often faces problems such as authenticity of the ratings given by customers and queries that leads to low accuracy of the recommendations. Implicit ratings play a vital role in providing refined ranking of products. Preference level of the customers are predicted based on collaborative filtering (CF) approach using implicit details and mining click stream paths of like-minded users. Extracting the similarity among products using sequential patterns improves the accuracy of ranking. Integrating these two approaches improves the recommendation quality. Based on the results of experiment carried out to compare the performance of CF, sequential path of products viewed and integration of the two, we conclude that integration of mentioned approaches is superior to the existing ones.

Introduction

A recommender system (RS) (Adomavicius and Tuzhilin 2005) is a tool that provides personalized services to its customers in e-commerce sites. These commercial websites have a large customer base and offer a vast variety of products. The customers hence have greater options to choose from several websites. This increase in competition has led the sites to use customer relationship management strategy to manage their customers. Elementary RS works on ratings obtained from user feedback and queries. However, recommendations do not turn to be accurate as ratings also depend on users' mood and time and hence are not reliable. Thus, instead of explicitly acquiring user ratings for specific products, implicit data (Kelly and Belkin 2001) represents customer's interest towards the target product.

A framework of calculating user preferences for products available on a commercial website is presented that is based on users' browsing and

CONTACT Shalini Gupta sgupta@arsd.du.ac.in ARSD College, University of Delhi, New Delhi 110021, India
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

sequential behavior which depicts their interests in favorable items. It finds solutions to three problems.

(1) Extracting implicit data based on user login details and time spent on product's webpage.
(2) Calculate preferences based on product viewed and applying recommendation techniques to calculate preferences.
(3) Preferences obtained above are refined by tracking sequential path users follow to reach the desired product.

In the first phase, preprocessing is applied on web log data to obtain implicit details of a user like time spent on viewing a product, number of views to a particular product made, whether products are viewed by searching or browsing, printing or bookmarking status for a particular product and basket/purchase status for a product. Second, classification techniques such as random forest (RF), artificial neural network (ANN) and gradient boosting (GB) are applied to calculate preferences for the products which are not purchased by the user and similarity measures like constraint Pearson correlation (CPC) (Shardanand and Maes 1995) and proximity–similarity–singularity (PSS) (Liu et al. 2014) are used to calculate preferences for the products that are not even viewed by grouping the similar users. Third, nearest neighbor's sequential paths are traced and target user's preference is refined based on calculating support for each product.

In this study, we derive a novel approach to refine the preferences of users for a particular product by categorizing the products as follows: products that are purchased, products that are placed in cart but not purchased, products that are viewed by corresponding users and products that are not even viewed. Purchased products will have highest preference over products that are basket placed but not purchased. Intuitively, products that are viewed but not basket placed has less probability of purchase. All other products have lowest priority as they are not even viewed. Also, user's sequential path is taken into consideration for refining the preferences obtained above.

In summary, the main contributions of this paper are twofold. Implicit data is extracted from user's browsing patterns, and based on these detail, user preference level for corresponding product is calculated. The priority levels obtained from above phase are refined based on sequential path followed by a user. Figure 1 describes the overall framework of the work carried out.

The entire paper contains various sections: Next section reviews related work regarding recommendation systems. Predicting preferences based on user's browsing behavior and sequential paths are defined in Proposed strategy section. Finally, experimental results and concluding remarks are presented in Experimental results and Conclusion sections, respectively.

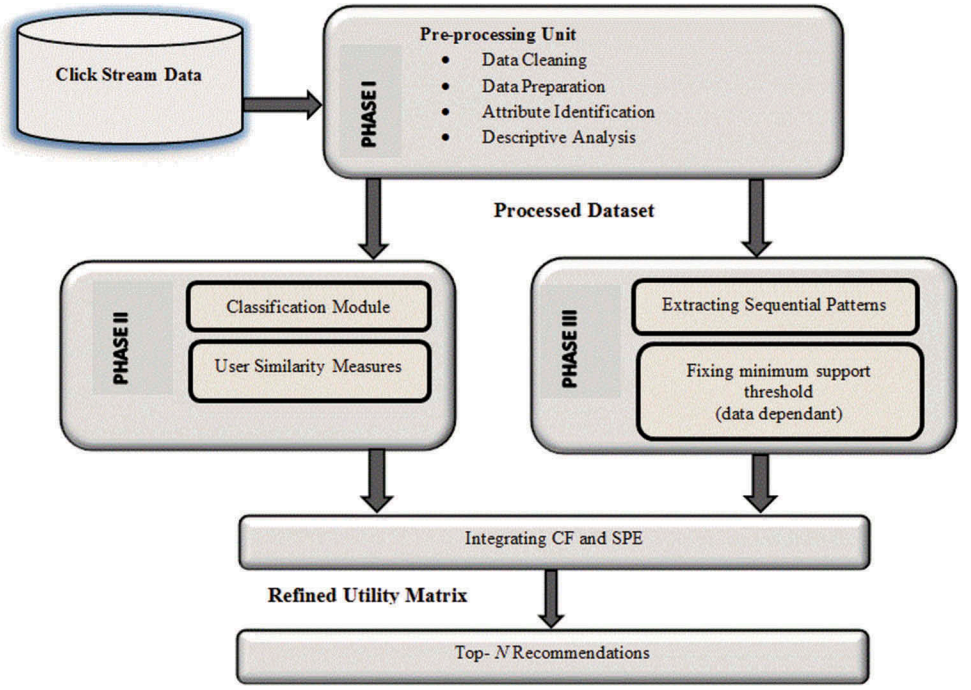

Figure 1. Overall framework.

Related Work

RS has been widely adopted in commercial sales and enterprises. Elementary techniques such as collaborative filtering (CF) (Adomavicius and Tuzhilin 2005; Si and Jin 2003; Yu et al. 2004), content-based filtering (CBF) (Lang 1995; Mooney and Roy 1999; Pazzani and Billsus 1997, 2007) and hybrid approaches (Liu et al. 2010; Wei, Yang, and Hsiao 2008) have been studied to resolve problems of RS such as novel user (or cold start) problem (Kim et al. 2010; Park and Chang 2009), new item (or first rater) problem (Lee and Kwon 2008) and sparse utility matrix problem (Kim et al. 2010; Lee and Olafsson 2009; Park and Chang 2009). An extension of CF technique namely cross-domain collaborative filtering adopts customer behavior from other related products to help recommendation in target category (Li, Yang, and Xue 2009; Pan et al. 2010; Sahebi and Brusilovsky 2013).

Conventional CF works well where users show their preferences for products in explicit manner such as ratings and queries. The like-minded users are grouped together for a particular product, and their ratings are taken as an average. The cold-start issue is one of the main challenges to existing CF-based RS (Cremonesi and Turrin 2009). CF also faces difficulties when only binary data (e.g. 'purchase' or 'no purchase') for a product are available (Hayes, Cunningham, and Smyth 2001). Many studies suggested methods that consider users navigational and behavioral patterns to predict

preferences (Kelly and Belkin 2001). Several authors presented case-studies by analyzing clickstream behavior of customers (Lee et al. 2001, 2000).

CBF systems construct an item profile by extracting set of features and build content-based user-profile from the items user purchased. Top-*N* items are recommended based on high similarity scores between user profile and item profile. These systems were used to recommend items such as books (Mooney and Roy 1999), net news (Lang 1995) and web pages (Pazzani and Billsus 1997). However, in CBF systems, it is difficult to obtain sufficient number of features to construct item profile (deficient feature problem) (Shardanand and Maes 1995). Also, the technique suffers from overspecialization problem (Adomavicius and Tuzhilin 2005) in which items recommended are same as that of purchased earlier.

Hybrid recommendation systems have been developed to overcome the limitations of CF, CBF and rule-based approaches (Liu, Lai, and Lee 2009; Liu et al. 2010). The technique works by combining the said approaches to eliminate insufficient features.

Problems associated with recommendation system have also been studied such as cold-start problem, first-rater problem and sparsity problem. Many solutions (Li et al. 2014, 2012) have been proposed to solve top-*N* recommendation problem. In summary, the proposed work differs from existing work in the following aspects:

- Top-*N* recommendations are generated on the basis of implicit data that is processed from click stream data obtained from big retailer e-commerce website.
- Preferences are calculated on the basis of data obtained by applying classification techniques, and similar users are grouped on the basis of similarity measures.
- These preference levels are refined by tracing the sequential path of similar users.

Proposed Strategy

Phase I: Pre-processing and Attribute Identification

The products in commercial website are generally categorized in a tree form. A hierarchical structure for product taxonomy which categorizes the products in different levels has root as database of all products. The first level of the hierarchy defines the main category of the product and the second level defines the subcategories. Last level depicts the item itself.

Attribute Identification

For the proposed RS, implicit data is extracted from clickstream variables. The attributes can be obtained by filtering the data from web log servers. All

Table 1. Description of attributes identified.

Attributes	Description	Variable type
Purchase	Product purchased or not	Binary (purchased = 1)
Cart placement	Product placed in the cart or not	Binary (cart placement = 1)
Category and subcategory ratio	No. of clicks of a particular category at particular level/total no. of clicks by the customer	Continuous
Visit mode	Searching = 1, browsing = 0	Binary
View count	No. of times a page is visited	Discrete
Duration of visit	Total time for which the customer visited the page/product.	Continuous

the click stream variables used in the proposed approach are explained in Table 1. For every customer, who clicked the webpage and viewed at least one product, the corresponding attributes are stored.

Descriptive Analysis

The data set represents six months of activities of a big e-commerce business in Europe selling all kinds of stuff such as electronics, clothes, toys and much more. Since most of the RS faces difficulty in recommendation as customers are involved in few transactions, the refined data set contains 5000 entries, with the clickstream data of 300 visitors and 100 products. These visitors have clicked more than 70 products where each product is allotted a separate webpage (Since webpages clicked can be repeated, the number of visits and reading time are summed up for the corresponding product.). The variables defined in Table 1 are to be analyzed for their impact on product purchased so that we can filter out only influencing variables. The probability that a customer purchased the product after placing it in cart can be inferred from Table 2. The value 0.84 (=813/969) shows that higher preference values will be associated with products that are placed in cart as compared to those not placed in the cart. Probability of purchase after the product was clicked through searching or browsing can be concluded from Table 3 (397/2467 = 0.161 and 416/2533 = 0.164, respectively)

Table 2. Probability of purchase for cart placement, visit mode and view count.

	Purchase = 1	Purchase = 0	Total
Cart placement			
Cart placement = 0	0	4031	4031
Cart placement = 1	813	156	969
Total	813	4187	5000
Visit mode			
Mode = 0 (browsing)	416	2117	2533
Mode = 1 (searching)	397	2070	2467
Total	813	4187	5000
View count			
1 view	77	866	943
2–4 views	206	2414	2620
5 or more views	530	907	1437
Total	813	4187	5000

Table 3. *t*-Test results for duration of visit and click ratios (5% significance level).

	N	Mean	StdDev	StdErr	Pr > *t*
Duration of visit					
Purchase = 0	4187	4.113	2.772	0.043	<0.0001
Purchase = 1	813	8.041	5.487	0.192	
Main category click ratio					
Purchase = 0	4187	0.293	0.103	0.002	0.424
Purchase = 1	813	0.296	0.104	0.004	
Subcategory click ratio					
Purchase = 0	4187	0.116	0.059	0.001	0.073
Purchase = 1	813	0.121	0.064	0.002	

again. These values imply that there are higher chances of purchase when the product is clicked through browsing as compared to when clicked through searching. In a similar fashion, the result shows that more view count leads to higher chances of product purchase (0.36 > 0.07 or 0.08).

Next, we use *t*-test to confirm three hypotheses. First, the difference in the mean value of visiting duration of purchased and not purchased products (with unequal variances) is found to be statistically significant at 5% significance level (as shown in Table 3) which indicates higher probability of purchase when visiting duration is higher. Second and third, for the differences in means click ratios of purchased and not purchased products of main category and subcategory (with equal and unequal variances, respectively), the hypothesis is not rejected at 5% significance level. It is concluded that the average of click ratios for main category and subcategory at 5% significance level is different significantly. So, it can be assumed that purchases are often made in customer's favorite main category and subcategory.

Phase II: Determining Preferences Using CF (PCF)

The attributes of the customer identified are used to predict the preference of products that have been clicked and CF is used to predict the preference for the products that have not been clicked by the customer at all. From these predicted preferences, products with highest value will be recommended to the user as shown in Figure 2. Products purchased by the customer will not be considered for recommendation assuming that products once purchased will have less probability to be purchased again. The steps followed in this phase are as follows:

(1) Preference for the products clicked/viewed by the customer is calculated using classification modules.
(2) Preference for the products that were not clicked by the customers is calculated with the help of the preferences of customers that are related to the target user using similarity measures such as CPC and PSS.

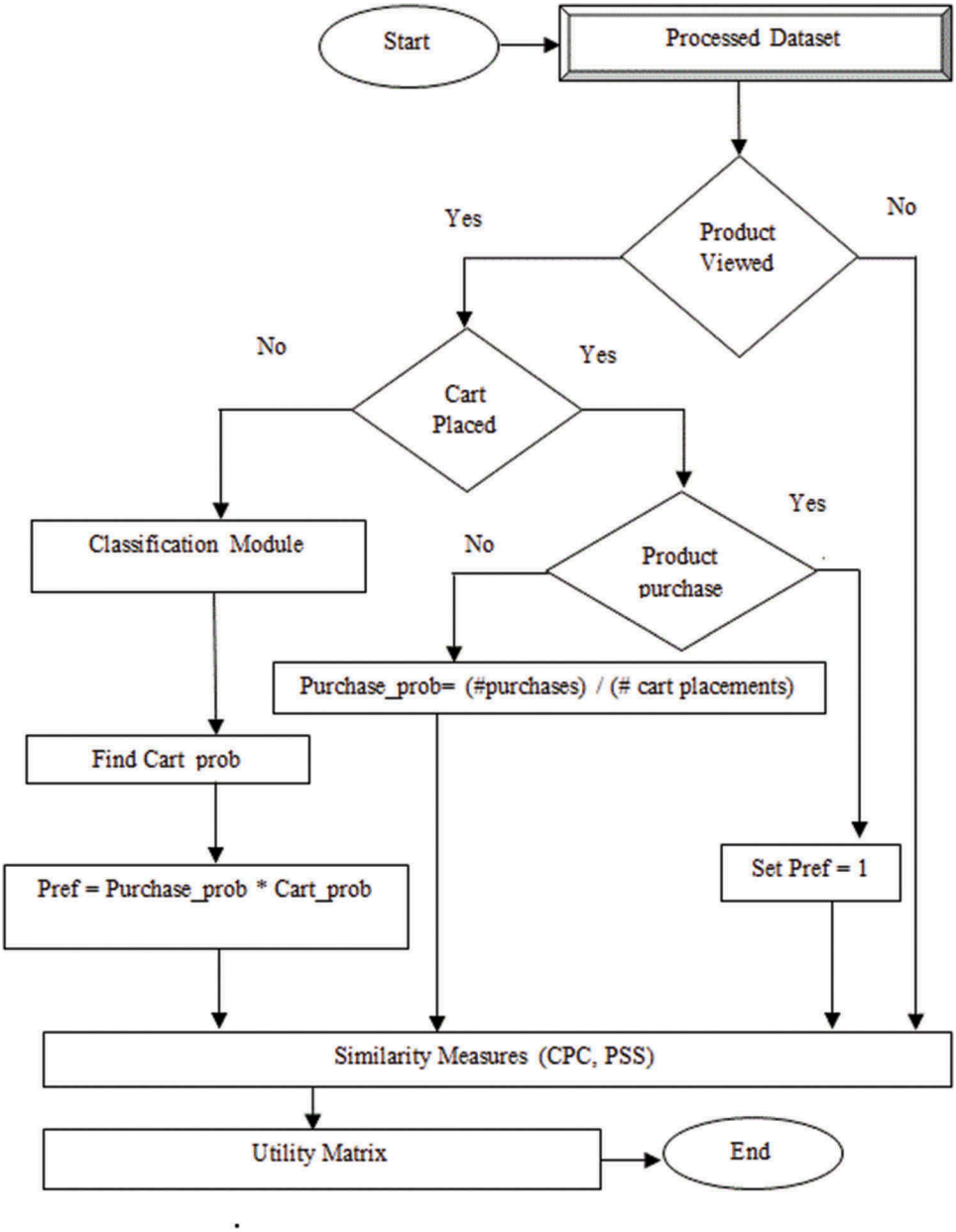

Figure 2. Calculation of preference levels in Phase II.

(3) Utility matrix is created which is to be further refined in Phase III by analyzing the path a user follows to reach the target product.

To calculate preferences for the products clicked by a customer, we divide them into three categories. First, the products that were purchased. For the products purchased, the preference is set to the highest possible value, i.e. 1. Second, the products not purchased but placed in the cart. The preference for

products that have been placed in the cart but not purchased is the probability of purchase after cart placement (Purchase_prob) that can be inferred from Table 3. Third, for the products that were not placed in the cart but were clicked, the preference cannot be calculated directly. So, the probability of cart placement (Cart_prob) of the clicked product by the customer is estimated and is multiplied by Purchase_prob to get the final preference for product purchase. To determine the probability of cart placement, some of the machine learning classifiers are used, which are supervised in nature. For the same, RF and GB are used.

Random Forest Classification (RF)

RF combines weak learners to form a strong learner. It considers decision tree as weak learner, and by combining multiple of them, it forms a strong learner. Thus, it is a collection of ensembles of decision trees, used for prediction on the basis of some predictor values. The result for prediction is found using Equation (1).

$$RF = \frac{1}{n}\sum_{i=1}^{n} DTResult(i) \tag{1}$$

The trees included are weak learners that combine to form a strong learner. For training purpose, random samples from the data set are used (with replacement), and for each random sample, a tree is built; each sample will have a left out that will be used for testing purpose. The final prediction is the average of predictions obtained from training with each random sample taken.

For the proposed approach, python's sklearn.ensemble library was used. The probability of cart placement has to be predicted. The predictors are duration of visit, view count, main and subcategory click ratios and visit mode. The classifier was trained in fourfolds, and accuracy of each fold was found out to be A1, A2, A3 and A4, respectively, giving average accuracy of the predictor as A. Gini index is used as the splitting condition for the tress generated.

Gradient Boosting Classification (GB)

GB combines weak learners but not necessarily decision trees. It iteratively generates models by reducing errors in preceding iterations, and the final prediction is calculated using sum of all iteration predictions. It is again an ensemble method which can use any weak learners as compared to RF where only trees are used.

In GB, multiple models are generated iteratively. In the first iteration, the procedure fits new model for providing the estimate of the response/target variable. In the next iterations, new models are generated in order to reduce the error in the previous iteration. The final prediction is the sum of the

predictions from each model as compared to average in the case of RF. The result for prediction is found using Equation (2). Sklearn.ensemble was used again for this approach.

$$GB = \frac{1}{n} \sum_{i=1}^{n} ModelResult(i)\,. \tag{2}$$

LR, DT and ANN are used as the weak leaners. Target variable and the predictors are the same as were in the case of RF. These classifiers were also trained in fourfolds, where accuracy of each fold was found out to be A1, A2, A3 and A4, respectively, giving average accuracy of the predictor as A. Results show that GB is a best classifier among others (as shown in Table 4).

Applying CF and Preparing Utility Matrix

After calculating the preference for the products clicked by the customer, the preference level for the products not clicked is calculated using CF. Preference levels of other customers that show similarity with the target customer are used to calculate the preference for products not clicked by the customer. We will calculate the similarity between users using PSS and CPC measures. Nearest neighbors of a target user will determine preference for products not clicked.

Proximity–Significance–Singularity (PSS)

Since users purchase small number of products as compared to total available products, we have preferences for a less number of products of a user which leads to less accurate similarity if other similarity measure such as CPC is used. The PSS measure assigns penalty to the bad similarity using its three factors. The proximity factor of PSS finds absolute difference between preferences of users and also checks for the agreement of preferences by giving penalty of disagreement. The significance factor of PSS finds the distance of preferences from median preference value as it is assumed that more the distant preferences are from median preference value, more would be the significance of the preferences. The singularity factor of PSS finds the difference of current two preferences with other preferences.

Table 4. Comparison of classifiers showing GB as the best classifier (on the basis of average).

	GB	RF	LR	ANN	DT
A1	0.9257	0.9161	0.9105	0.9041	0.8625
A2	0.9184	0.912	0.9048	0.8992	0.8496
A3	0.908	0.904	0.8976	0.8968	0.8488
A4	0.9167	0.8967	0.8967	0.8975	0.8439
Average accuracy **(A)**	**0.9172**	0.9072	0.9024	0.8994	0.8512

The similarity among users is calculated using Equation (3), where $PSS_{pref(a,p),\ pref(b,p)}$ is similarity between user a and user b for the commonly clicked product p that belongs to I-dimensional preference vector scale.

$$Sim_{a,b} = \sum_{p \in I} PSS_{pref(a,p),\ pref(b,p)}. \tag{3}$$

PSS measure can be calculated using Equation (4).

$$PSS_{pref(a,p),pref(b,p)} = Prx_{(pref(a,p),pref(b,p))} \times Sgn_{pref(a,p),pref(b,p)} \times Sngl_{pref(a,p),pref(b,p)} \tag{4}$$

where $Prx_{(pref(a,p),pref(b,p))}$ is proximity between user a and user b for product p.
$Sgn_{pref(a,p),pref(b,p)}$ is significance of user a and user b for product p.
$Sngl_{pref(a,p),pref(b,p)}$ is singularity of user a and user b for product p.
The three factors of PSS are given by Equations (5)–(7).

$$Prx_{(\mathrm{pref(a,p),pref(b,p)})} = 1 - \frac{1}{1 + exp(-|pref(a,p) - pref(b,p)|)}. \tag{5}$$

$$Sgn_{pref(a,p),pref(b,p)} = \frac{1}{1 + exp(-|pref(a,p) - pref_{med}|^{*}|pref(b,p) - pref_{med}|)} \tag{6}$$

$$Sngl_{pref(a,p),pref(b,p)} = 1 - \frac{1}{1 + exp\left(-\left|\left(\frac{pref(a,p)+pref(b,p)}{2}\right) - \mu(p)\right|\right)} \tag{7}$$

Constrained Pearson Correlation (CPC)

Another proximity measure that finds similarity among user a and user b is CPC that is based on preferences for commonly clicked products (as shown in Equation (8)).

$$Sim_{a,b} = \frac{\sum_{p \in I}(pref(a,p) - v)(pref(b,p) - v)}{\sqrt{\sum_{p \in I}(pref(a,p) - v)^2 \sum_{p \in I}(pref(b,p) - v)^2}} \tag{8}$$

where v is the midpoint of the scale. Assuming the range of preference between 0 to 1, we consider the midpoint v as 0.5. $pref(a,p)$ is the preference value of user a for product p. $pref_{med}$ is the median of preferences. $\mu(p)$ is the average of preferences for product p. The algorithm for the proposed approach is shown in Figure 3.

Input:

- Processed Dataset D consisting of
  - Attributes (predictors)
    - Visit Mode
    - View Count
    - Duration of visit
    - Main_category_Click_Ratio
    - Sub_Category_Click_Ratio
  - Attributes (non − predictors)
    - Cust_ID
    - Prod_ID
    - Cart_placement
    - Purchase_status

Function:

```
Purchase_prob = # of purchases / # of cart_placement
Model = Train (classifier, D)
For each transaction in D:
        Cust = transaction[Cust_ID]
        Prod = transaction[Prod_ID]
        If transaction [Purchase_status] == 1:
                Preference [Cust_ID][Prod_ID] = 1;
        Elif transaction[Cart_placement] == 1:
                Preference[Cust_ID][Prod_ID] = Purchase_prob;
        Else :
                Cart_prob = pred_prob(Model, transaction[predictors]);
                Preference[Cust_ID][Prod_ID] = Cart_prob * Purchase_prob;
End For
Similarity1 = PSS (Preference)
Similarity2 = CPC (Preference)
Similarity = max (Similarity1 , Similarity2 )
Final_Preference = CF (Preference, Similarity)
For each row in Final_Preference:
        Recommend.append(top_N(Preference))
End For
```

Output:

Return top-N Recommendations

Figure 3. Determination of preference level algorithm.

Phase III: Determining Preferences Using Sequential Patterns (PSP)

In this phase, the preference of users for corresponding product is predicted based on the sequential path they follow to reach the target product. The sequential paths of the users that are neighbors of target user are considered in this method. The path is a sequence of product IDs, arranged by their viewed time. The path traced to reach each product clicked is derived from minimum to maximum length and is compared to the path of target user. The length obtained is compared with minimum support threshold, and

permissible values are added to get overall preference for the product. The preference of user a for product p is calculated using Equation (9).

$$Pref_{a,p} = \sum_{s \in SEQ} Sup_s^p \tag{9}$$

where SEQ denotes set of subsequences of user a and Sup_s^p denotes support of product p from a subsequence s. After normalizing the preferences obtained from Phases II and III, target user a's final preference is calculated using Equation (10).

$$Pref_{a,p} = \propto \times PCF(a,p) + (1 - \propto) \times PSP(a,p) \tag{10}$$

An Illustration with Example

Suppose target user a (having sequential path <P1> <P3> <P2>) is having four-nearest neighbors with sequential paths given as in Table 5.

We will find the preference of each product by traversing the sequential behavior of all the neighbors. Now by finding the subsets of sequential pattern for the target user a in left to right manner, i.e. <P1>, <P3>, <P2>, <P1><P3>, <P1><P2>, <P3><P2>, <P1><P3><P2>, we will find the probability of each product to be clicked following the path in the subsets. Considering the minimum support as 0.5 for sequential pattern mining, following subsets are obtained from nearest neighbor's sequential path. Since the path followed is sequential, pairs are generated in left to right direction. Table 6 shows the aggregate preference of target user a for the given products.

Experimental Results

Determination of Preference Levels

Using the preference values so obtained, non-purchased products with highest preferences by the customer (i.e. preference <1) are recommended. Recommended items varied in the study by a factor of 5.

To evaluate the proposed RS, few products having value 1 are hidden from the preference level matrix of customers and products. Preference values of products for the customers having hidden products will be calculated using proposed RS, and top-N products will be recommended for those customers.

Table 5. Sequential path of neighbors of target user a.

Neighbors of user a	Sequence of products clicked
User 1	<P1> <P3> <P4>
User 2	<P1> <P3> <P4> <P5>
User 3	<P2> <P3> <P5>
User 4	<P1> <P2> <P3>

Table 6. Preference level of target user *a* for corresponding products (minimum support threshold = 0.5).

	Product 1	Product 2	Product 3	Product 4	Product 5
Length 1	**<P1>: 0.75**	**<P2>: 0.5**	**<P3>: 1.0**	**<P4>: 0.5**	**<P5>: 0.5**
Length 2	–	<P1><P2>: 0.25	**<P1><P3>: 0.75**	**<P3><P4>: 0.5**	<P4><P5>: 0.25
			<P2><P3>: 0.5	**<P1><P4>: 0.5**	**<P3><P5>: 0.5**
					<P1><P5>: 0.25
					<P2><P5>: 0.25
Length 3	–	–	<P1><P2><P3>:0.25	**<P1><P3><P4>: 0.5**	<P3><P4><P5>: 0.25
					<P2><P3><P5>: 0.25
					<P1><P3><P5>: 0.25
Length 4	–	–	–	–	<P1><P3><P4><P5>:0.25
$Pref_{user\ a}$	0.75	0.5	2.25	2.0	1.0

Performance could be decided by checking for the hidden products in the top-*N* products so recommended. The evaluation could be measured using recall and precision measures which are given by Equations (11)–(12).

$$recall = \frac{\sum_{i \in X} |\mathrm{Hid(i)} \cap \mathrm{Top\_N(i)}|}{\sum_{i \in X} \mathrm{Hid(i)}} \tag{11}$$

$$precision = \frac{\sum_{i \in X} |\mathrm{Hid(i)} \cap \mathrm{Top\_N(i)}|}{\mathrm{N}.|\mathrm{X}|} \tag{12}$$

where *Hid*(*i*) is the *i*th customer hidden products.

N is the recommended products to each customer.

Top_N(i) is the Top *N* products recommended to *i*th customer.

X is the no. of customers with hidden products.

Since these measures inversely get affected by change in attributes such as *N*, a combination of these measures *F1* (Sarwar et al. 2000) is used for evaluation purpose given by Equation (13). *F1* measure with high value indicates better performance of proposed RS.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{13}$$

The numbers of recommended items are set to 5, 10, 15, 20 and 25. It is noted from Figure 4(a) that *F1* gives highest result when *GB* is used as a classifier for calculation of preference levels. As shown in Figure 4(b)–(d), PSS outperformed CPC in all cases. It can be inferred that PSS is more appropriate as a similarity measure than CPC. The best accuracy of PSS is obtained when recommended items are 25.

Items recommended are different based on the minimum support selected. Experiments for the proposed work are carried out considering minimum support as 1% (PSP_1), 2% (PSP_2) and 3% (PSP_3). Figure 5 depicts that PSP_1 outperformed PSP_2 and PSP_3, i.e. to extract sequential patterns from click stream data when minimum support is considered as 1%. After determining the preference levels based on PCF and PSP methods, the accuracy of final

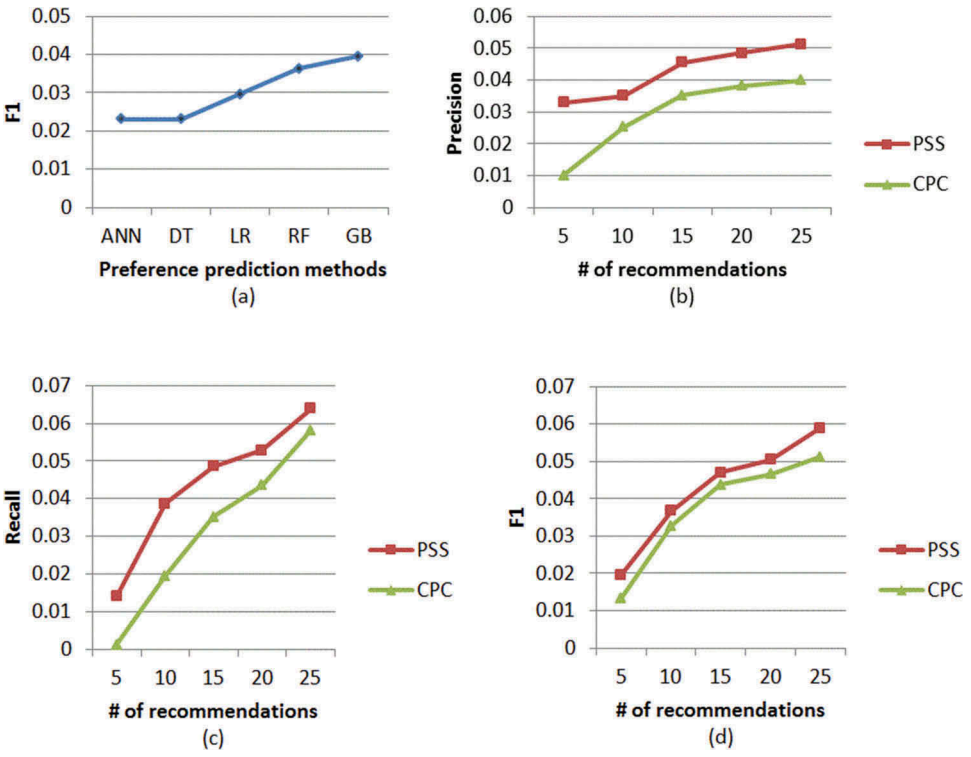

Figure 4. Comparison of precision, recall and *F1* when similarity measures used are PSS and PCC (nearest neighbors considered as 4).

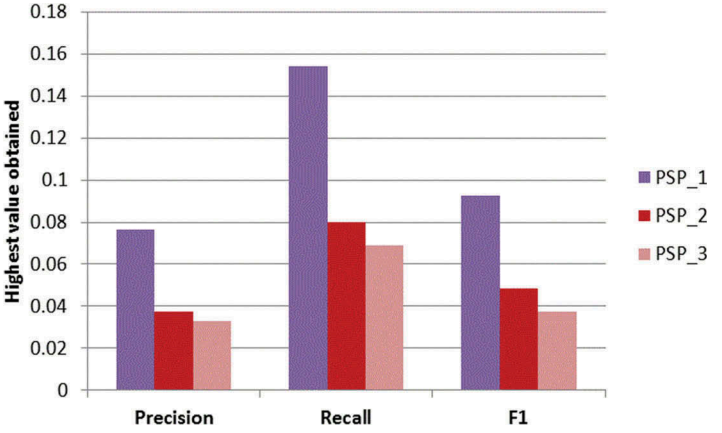

Figure 5. Comparison of precision, recall and *F1* using minimum support as 1%, 2% and 3%.

recommendation is calculated based on assigning integration weights (α) to each phase. Figure 6 shows that precision, recall and *F1* increase when α increases from 0.1 to 0.2 but decreases when it is approaching towards 1.

Therefore, the algorithm gives the best result when PCF technique and PSP experiment are assigned the weights as 0.2 and 0.8, respectively.

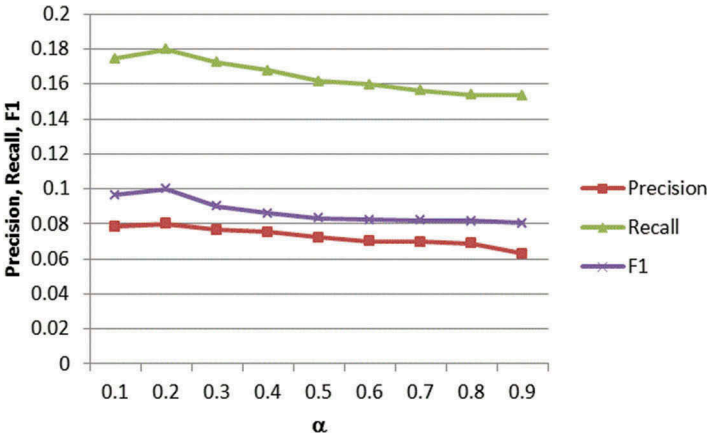

Figure 6. Adjustment of integration weight.

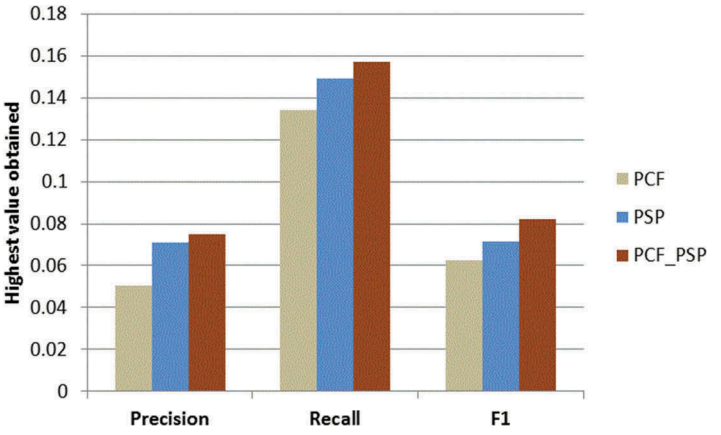

Figure 7. Comparison of information retrieval methods among three approaches.

As shown in Figure 7, PCF_PSP outperformed PCF and PSP in all information retrieval methods. This illustrates that sequential path of similar users along with collaborative preferences improves the accuracy of recommendation.

Conclusion

In this paper, the proposed approach utilizes the clickstream data and predicts the preferences of the products clicked by the customer using machine learning algorithms (GB and RFs). For the products not clicked, CF is used to predict the preferences considering PSS measure to find similarity between users. These preferences are refined by tracing the sequential path of similar users in left to right direction. Products that appear in the sequence that fulfill minimum support threshold are weighted high as compared to rest. In

the experiment carried out, minimum support is taken as 1%. Both the methods are assigned integrated weights ($\alpha = 0.2$ and $1\text{-}\alpha = 0.8$ resp.), so that top-*N* preference products are recommended to the user.

For the analysis and evaluation of the RS, *F1* values are analyzed. From the analysis, we observed that the two new classifiers used outperform weak learners (DT, ANN and LR). The proposed RS outperforms above two approaches in terms of precision, recall and *F1*. The mining approach used covers only those customers who had viewed more than 70 products. So the values are subject to verification for huge real-world data where a large set of customers view only a chunk of products.

References

Adomavicius, G., and A. Tuzhilin. 2005. Toward the next generation of recommender system: A survey of the state of the art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17 (6):734–49.

Cremonesi, P., and R. Turrin. 2009. Analysis of cold-start recommendations in IPTV systems. Proceedings of the 3rd ACM Conference on Recommender Systems, New York, USA, 233–36.

Hayes, C., P. Cunningham, and B. Smyth. 2001. A case-based reasoning view of automated collaborative filtering. Proceedings of the 4th international conference on case-based reasoning, Berlin, Heidelberg, 234–48. Springer.

Kelly, D., and N. J. Belkin. 2001. Reading time, scrolling, and interaction: Exploring implicit sources of user preferences for relevance feedback. Proceedings of the 24th annual ACM SIGIR conference on research and development in information retrieval, New Orleans, LA, 408–09. ACM.

Kim, H. N., A. T. Ji, I. Ha, and G. S. Jo. 2010. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications* 9 (1):73–83. doi:10.1016/j.elerap.2009.08.004.

Lang, K. 1995. Newsweeder: Learning to filter netnews. Proceedings of the 12th international conference on Machine Learning, 331–39.

Lee, J., M. Podlaseck, E. Schonberg, and R. Hoch. 2001. Visualization and analysis of click stream data of online stores for understanding web merchandising. *Data Mining and Knowledge Discovery* 5 (1-2):59–84.

Lee, J., M. Podlaseck, E. Schonberg, R. Hoch, and S. Gomory. 2000. Understanding merchandising effectiveness of online stores. *Electronic Markets* 10 (1):20–28. doi:10.1080/10196780050033944.

Lee, J. S., and S. Olafsson. 2009. Two-way cooperative prediction for collaborative filtering recommendations. *Expert Systems with Applications* 36 (3):5353–61. doi:10.1016/j.eswa.2008.06.106.

Lee, K. C., and S. Kwon. 2008. Online shopping recommendation mechanism and its influence on consumer decisions and behaviors: A causal map approach. *Expert Systems with Applications* 35 (4):1567–74. doi:10.1016/j.eswa.2007.08.109.

Li, B., Q. Yang, and X. Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, Canada, 617–24. ACM.

Li, D., Q. Lv, L. Shang, and N. Gu. 2014. Item-based top-N recommendation resilient to aggregated information revelation. *Knowledge Based Systems* 67:290–304. doi:10.1016/j.knosys.2014.04.038.

Li, D., Q. Lv, X. Xie, L. Shang, H. Xia, T. Lu, and N. Gu. 2012. Interest-based real-time content recommendation in online social communities. *Knowledge Based Systems* 28:1–12. doi:10.1016/j.knosys.2011.09.019.

Liu, D. R., C. H. Lai, and W. J. Lee. 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences* 179 (20):3505–19. doi:10.1016/j.ins.2009.06.004.

Liu, H., Z. Hu, A. Mian, H. Tian, and X. Zhu. 2014. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems* 56:156–66. doi:10.1016/j.knosys.2013.11.006.

Liu, Z., W. Qu, H. Li, and C. Xie. 2010. A hybrid collaborative filtering recommendation mechanism for P2P networks. *Future Generation Computer Systems* 26 (8):1409–17. doi:10.1016/j.future.2010.04.002.

Mooney, R. J., and L. Roy. 1999. Content-based book recommending using learning for text categorization. Proceedings of the fifth ACM conference on Digital libraries, Berkeley, CA, 195–204. ACM.

Pan, W., E. W. Xiang, N. N. Liu, and Q. Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. Proceedings of the 24th AAAI Conference on Artificial Intelligence, 230–35.

Park, Y. J., and K. N. Chang. 2009. Individual and group behavior-based customer profile model for personalized product recommendation. *Expert Systems with Applications* 36 (2):1932–39. doi:10.1016/j.eswa.2007.12.034.

Pazzani, M., and D. Billsus. 1997. Learning and revising user profile: The identification of interesting web sites. *Machine Learning* 27 (3):313–31. doi:10.1023/A:1007369909943.

Pazzani, M. J., and D. Billsus. 2007. Content-based recommendation systems. The Adaptive Web: Lecture Notes in Computer Science, 325–41. Springer Berlin Heidelberg.

Sahebi, S., and P. Brusilovsky. 2013. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. International conference on User Modeling, Adaptation, and Personalization, Rome, Italy, 289–95. Springer Berlin Heidelberg.

Sarwar, B., G. Karypis, J. Konstan, and J. Riedl. 2000. Analysis of recommendation algorithms for e-commerce. Proceedings of the 2nd ACM conference on Electronic commerce, Minneapolis, MN, USA, 158–67. ACM.

Shardanand, U., and P. Maes. 1995. Social information filtering algorithms for automating "word of mouth". Proceedings of the SIGCHI Conference on Human factors in computing systems, Denver, CO, USA, 210–17. ACM.

Si, L., and R. Jin. 2003. Flexible mixture model for collaborative filtering. Proceedings of 20th International Conference on Machine Learning, Washington, D.C. 704–11.

Wei, C. P., C. S. Yang, and H. W. Hsiao. 2008. A collaborative filtering-based approach to personalized document clustering. *Decision Support Systems* 45 (3):413–28. doi:10.1016/j.dss.2007.05.008.

Yu, K., A. Schwaighofer, V. Tresp, X. Xu, and H. P. Kriegel. 2004. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering* 16 (1):56–69. doi:10.1109/TKDE.2004.1264822.