Improvising Purchase Prediction using hybrid approach utilizing Apache Mahout with C45. And Naive Bayes classifier

[1]Pooja Pawar, [2]S.G. Tuppad

[1]ME Student, [2]Professor
Department of CSE, MSSCET Jalna,

***Abstract*—Ecommerce has grown rapidly over the years. Many people are using popular channels to buy goods and services on the internet. It therefore becomes very important for shopping sites to accurately predict what customers want to buy to increase sales or increase customer satisfaction. Traditional algorithms like joint filtering are extremely popular in predicting user settings in the book, guide, or music area. But they face issues where rating data is sparse or not available in the shopping domain. Compared to the number of ratings on an ecommerce shopping site, the volume of many user click data is sufficient for the user's liking. So in this article, we propose a prediction model based on probability statistics that use user click behavior data**

Index Terms— E-commerce Purchase Prediction; Clicking Behavior Data; Probability Statistics

---

I. INTRODUCTION

Due to the growing popularity of online shopping market today, competition between eCommerce companies to attract more users is hugely intense. For this reason, you need to understand and describe the user's online shopping behavior. It has been reported that Amazon.com generates about 35% revenue through referral tools. [9] This means that modeling and predicting future user acquisition is paramount. Successful Ecommerce In order to accurately predict the future purchase of a user, past purchases must be recorded with a clear response. Especially clear responses like ratings and reviews.

User-specific settings are clearly the most valuable user feedback. However, most users do not buy products on a regular basis and even if they do. But it does not give a clear suggestion about the product purchased. For this reason, ecommerce companies should rely on implied implicit user suggestions, such as past purchase records, when creating their systems.

Many studies have focused on the use of implied user feedback. To set the user preferences Hu and the faculty [3] introduced the concept of confidence that non-purchase items will be reduced. Rendle et al. [11] offers a learning-to-rank pair, called the Bayesian Personalized Ranking (BPR), which each user will consider purchasing. On all items not purchased are considered Negative comments (such as All Missing) are negative assumptions (AMAN) [10]).

In addition, the problem of data loss due to recording. The purchase of the user is often rare. To overcome the above limitation, we take advantage of the user's 'past click' logs overlooked by previous tasks in conjunction with A user's purchase log. Clicking log is another type of user implied. The comments are ridiculous. Implicitly, the user is precise, while not being selected to purchase. Of all the clicks, the list continues. Disclosure of user interest in the user group since the purchase option has been selected. In that very clickable list, we expect a click. The record helps relieve AMAN assumptions when combined with purchase records.

Also, because in operation, clicking Save often exceeds the volume of purchase records, we hope that user click logs will help to diagnose the problem of data fragmentation. In this article, we define a pairwise relationship between items. Strive to remove the limitations of existing work and propose a new approach called P3S, which represents a dual model. The harmony between disorganized sets, which allows users to 'click' together with their precise purchase logs, eyes, purchase notes, and user logs, so we split the items. Three sets of disorganized dresses: 1) Buy items 2) Click but did not buy. items, and 3) non-clicked items, and define new pairwise relationships among these item sets with respect to users.

Reflecting on these relationships as a way of learning in pairs, we show that the accuracy of predictions of future user acquisition will increase dramatically. The results in two sets of real-world data show that the way we deliver is better than the most advanced way to predict users' future ecommerce purchases.

II. RELATED WORK

Wherever Times is specified, In this section, we review the study that is directly relevant to us, namely the recommender system, using an implicit feedback package, and a method for modeling user behavior. Recommender systems with implied suggestions. Thanks to its abundance compared to the obvious suggestion, research to create a recommender system based on user feedback implied immense interest. Suggested user comments include clicking on the news site, listening history, checking behavior, and channel customization events. In particular, Hu et al. [3] and Pan et al. [10] proposed the appropriate weighting for item suggestions using implied feedback using the confidence matrix to discriminate. The influence of purchased items (purchased) and items that have not been inspected.

Market basket analysis [29, 31] examines groups of items purchased in supermarkets or other outlets to identify purchase patterns, and discrete choice models [16, 30] predict which products a specific customer is likely to select from a candidate product set. Certain studies abstract the prediction of purchase behavior into a classification task [7, 10] that requires customers' demographic features, such as age, gender, education and occupation. Unlike traditional businesses, firms in the e-commerce context find it difficult to obtain information about customer demography or family background because these data are usually regarded as private [19]. Rather, it is more convenient to access customer reviews, product ratings and visiting tracks. Therefore, the methods and algorithms used to predict customer purchase behavior in the traditional business context must be modified for e-commerce. To meet the challenge of predicting purchase behavior in the e-commerce context, we first examine the purchasing decision process of customers.

Guo and Barnes [8] propose a three-stage purchasing decision process in the e-commerce context, which is illustrated in Fig. 1. The first stage, problem/motivations1 recognition, captures consumers' perceptions of how products may help them to bridge the gap between the desired and actual states. During the second stage, a customer must seek information about product performance or other criteria and must evaluate product alternatives based on price, brand and other attributes.

Recommender system technology has been widely adopted by e-commerce websites. This technology not only provides appropriate recommendations to users but also yields substantial profits for the service provider. Nevertheless, we believe that traditional recommendation algorithms cannot effectively predict purchasing behavior in the e-commerce context, for three reasons: (1) Recommender systems [12, 17, 32, 33] predict the items most likely to interest a customer either by exploiting product ratings by other customers with similar tastes [collaborative filtering (CF)] or by using past product ratings by the target customer (content-based recommendations). However, the rating predicted by a recommender system for a candidate product only indicates the impression the customer will likely have of that product, i.e., it predicts a ''like'' signal. This is a far cry from the goal of predicting purchase behavior. According to our experiments, the use of CF alone to predict customer purchase behavior leads to a poor performance. Reference [28] notes that a consumer usually makes purchasing decisions based on a product's marginal net utility. A rational consumer chooses to purchase the product that maximizes total net utility.

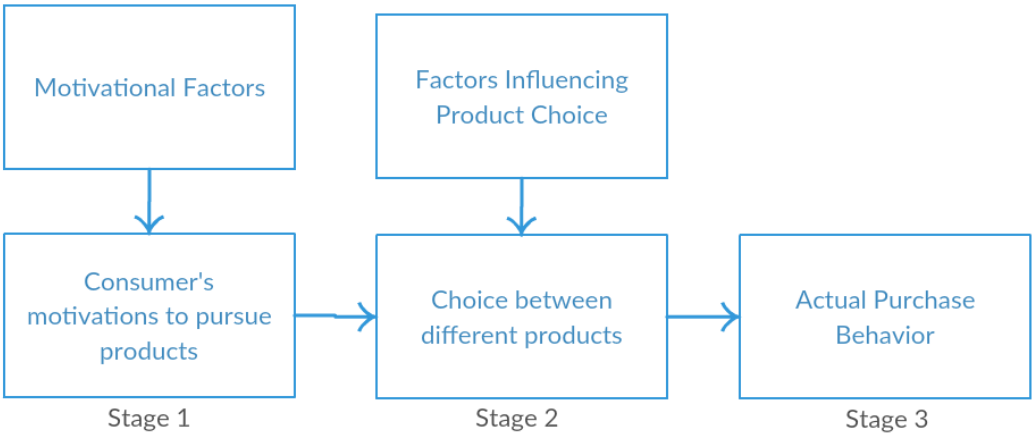

Fig. 1 Consumer behavior and the purchasing decision in the e-commerce context

2) Many works on recommender systems focus on the user–item matrix and employ tensor factorization. However, those studies only consider associations among users and items and ignore rich product features.2 We think that side information plays an important role in the second stage of the purchasing decision process depicted in Fig. 1.

(3) Recommender systems usually do not address the first stage of the purchasing decision process shown in Fig. 1. Rather, recommender systems address only a single component of the second stage of the purchasing decision process. In addition, some recommender systems aim to increase product awareness [4] and thus consider freshness and novelty in product recommendations [18]. Such systems cannot be employed to perform the predictive task.

In this study, we propose a two-stage predictive framework. First, customer's motivations are investigated. However, not all motivations that affect purchases of real products can be investigated in the e-commerce; one such example is Maslow's psychological and safety needs. We exploit the associations among products to predict a customer's motivations.

The second step of the framework addresses the numerous factors that affect customer decision making, such as price [25], brand preference, economic needs [23], etc. We seek to integrate side information about products into the predictive task by learning customers' preferences for particular product features. Furthermore, we leverage these preferences to select a collection of candidate products based on customer's motivations. When a customer submits one purchased product into the predictive framework, this framework can return the top n products most likely to be purchased by that customer in the future..

III. Overview

The Recommender framework helps clients to discover and evaluate their investments. The Recommender system can utilize data mining strategies to provide guidance based on knowledge gained from the activity and quality of the customer. The Miner's Guide to online guidance systems on the web. Internet use generally consists of three steps: data processing, pre-detection, pattern generation, and hinting. The process of data processing and pattern detection is performed offline and instructions are generated online. Previous data processing involved converting log files to web access and user profiles in a form suitable for the system. Includes pattern validation using data mining techniques such as custom mining groupings or mining rules. Finally, the detected pattern is used to generate instructions that provide links or custom information to the user. Each column has a subset of highly relevant attributes.

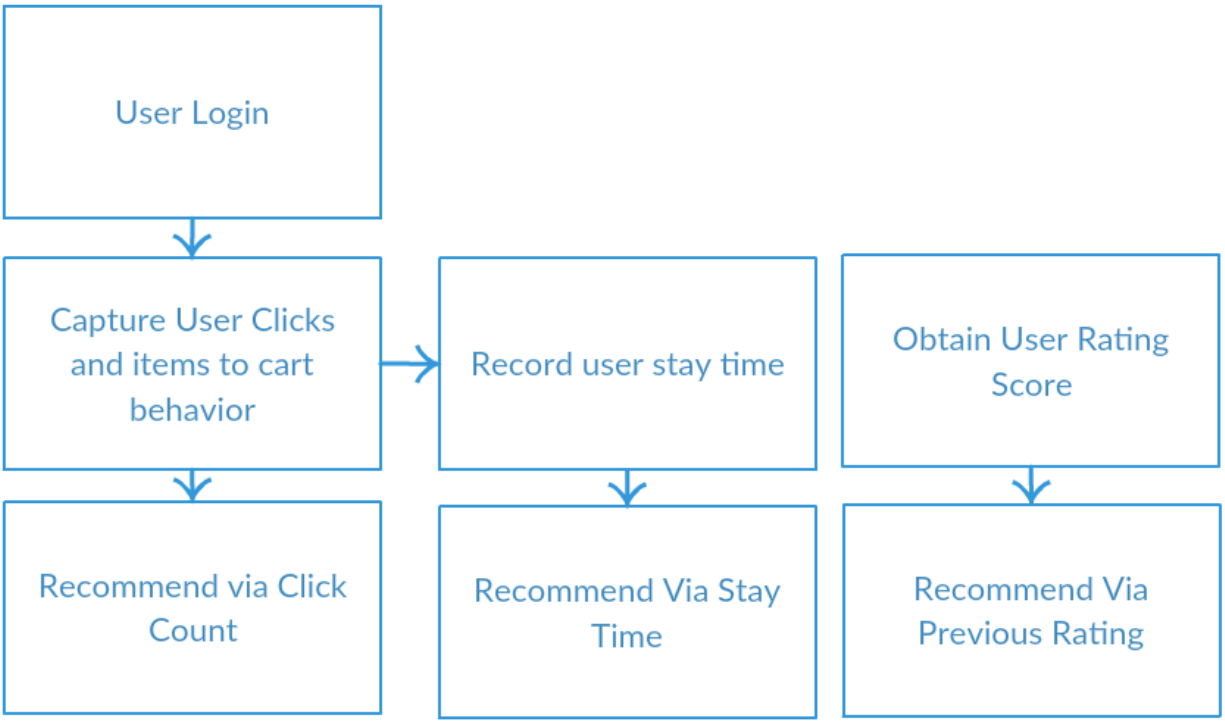

Figure 2 System Flow Diagram

The improved system architecture above the instructions involves additional integrated user data. As a result, users with common behavior are grouped first based on their clicks and add to cart behavior and then grouped into groups. Further based upon their necessity users are recommended products based upon system prediction. New users are ranked first in a group, and then use the corresponding group format to set current user and other similar user guidance in the top group segments.

Cold Start Problem

Suggested tools that use common filtering suggest each element. (The mobile phone advertised on site), depending on the user's actions. The more user actions are taking place, the more interested users are, and what other elements are more similar. Over time, the system will be able to give you more accurate advice. However, this has caused confusion and difficulty in organizing their site and suggestion tools. Although newer ads are most relevant, suggestion systems are less likely to recommend users than previous articles. But do not let old ads dominate the introduction process.

Solution

Filtering based on content is the method that answers this question. Our system first uses the metadata of new products when creating recommendations, while the visitor's action is secondary for a certain period of time. In addition, we can identify visitors who are only there to navigate and certain visitors who know what they are looking for. For example, if someone clicks on everything from phone cases to real estate in a short period of time, the system will assume that they are only there to navigate and will not use their click history for recommendations. When it comes to investigating the phenomenon of cold start, this is just the tip of the iceberg. Each recommendation solution has a different method to cope with it, and after overcoming the hard cold start, the real work of the engine begins.

However, product to product recommendations will persist in any case, regardless of whether users block cookies or not.

3.1.1. Offline Phase

This process consists of two main modules: data processing and knowledge base of pre-processing products. In the process, I started with offline pre-processing basics. Web-Access-Log This includes splitting the client session and entering important data in the database.

1) Data Processing: At this stage, the source files are formatted to find the Web access range. Web server logs are generated throughout the user's web server access. There are different types of web logs based on different server parameters. These records include information such as URLs, IP addresses, clients, etc. Pre-processing features, such as session cleanup data, are performed prior to using Web mining algorithms on web server logs. Grouping is also carried out in the process. Here k refers to the clustering algorithm used. In the k-guide system, means can be used in the pre-processing process for identifying groups of users appears to be similar. Used to collect user profiles.

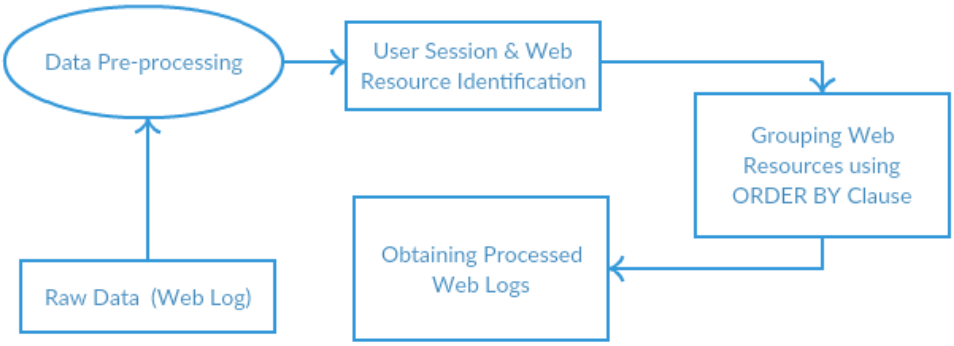

Figure 3 Data Pre Processing & Processed Web Logs

2) Knowledge base: After the data processing, product data information is merged with the user session data extracted from the record. These features include brand prices, user details, and transactions. It performs in a table in an advanced database system.

3.1.2. Online phase Architecture

In this session, when the user logs on to the server, the instructions are controlled with the knowledge base for the above transaction with the user. A list of recommended products is created based on the user's previous history and the type of group the user is a member of.

1) Creating Suggestions: An important system utility is to create recommendations using some refining parameters such as brand valuation and other customizable parameters to get a certain set of support values defined elements of the database.

Historical logs will also allow us to identify the location of user, the kind of products that user is looking for, adding to cart or wish list.

Such information is termed as meta data and is quite useful for system to overcome cold start problem.

Based on purchase history and historical logs i.e. session logs, we will classify user's session and proceed towards recommendation using following classifiers:

1) C 4.5
2) Naive Bayes

C45 Algorithm

J48 or C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

Algorithm

C4.5 creates few decision trees from the set of training data and it is likely to create in the same way as ID3, using the eventual concept of Entropy of information. The training data is a set of samples already classified. Each sample Consists of a p-dimensional vector, where they represent attributes or characteristics of the sample, as well as the class in which it falls.

At each node of the tree, C4.5 chooses the data attribute that most effectively divides its set of samples into enriched subsets in one class or the other. The splitting criterion is the gain of standardized information (difference of entropy). The attribute with the highest standardized information gain is chosen to make the decision. The algorithm C4.5 then returns to the smaller sublists.

This algorithm has some basic cases.

• All samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree by saying to choose this class.

• None of the features provide information gain. In this case, C4.5 creates a higher decision node in the tree using the expected value of the class.

• Previously invisible class instance encountered. Again, C4.5 creates a decision node higher in the tree using the expected value.

The C5.0 algorithm is an extension of the C4.5 algorithm which is also the extension of ID3. This is the classification algorithm that applies in the large dataset. It's better than C4.5 on speed, memory and efficiency. Model C5.0 works by dividing the sample according to the field that provides the maximum information gain. Model C5.0 can divide the samples on the basis of the larger information gain field. The sample subset that is obtained from the first division will be divided thereafter. The process will continue until the subset of samples can not be split and is usually in a different field. Finally, examine the lowest level fraction, the subsets of samples that have no remarkable contribution to the model will be rejected. C5.0 easily handles the multi-value attribute and the missing attribute of the dataset [3].

Rule sets: An important feature of is to generate classifiers called rule sets that consist of unordered collections of (relatively) simple if-then rules. The Rule sets option causes classifiers to be expressed as rule sets rather than decision trees.

Each rule consists of:

• A rule number - this is quite arbitrary and serves only to identify the rule.

• Statistics (n, lift x) or (n / m, lift x) that summarize the performance of the rule. Similarly to a sheet, n is the number of training cases covered by the rule and m, if it appears, shows how many of them do not belong to the class predicted by the rule. The precision of the rule is estimated by the Laplace ratio (n-m + 1) / (n + 2). The elevator x is the result of dividing the estimated accuracy of the rule by the relative frequency of the predicted class in the drive assembly.

• One or more conditions that must be met for the rule to apply.

• A class predicted by the rule.

• A value between 0 and 1 that indicates the confidence with which this prediction is made.

When a rule like this is used to classify a case, it may happen that several of the rules are applicable (ie, all of their conditions are satisfied). If the applicable rules provide for different classes, there is an implicit conflict that could be solved in several ways: for example, one might think that the rule is the greatest confidence, or one might try to aggregate the rule predictions to arrive at To a verdict. Rule sets are usually easier to understand than trees because each rule describes a specific context associated with a class. In addition, a rule set generated from a tree usually has fewer rules than the tree has leaves, another more for comprehensibility. Another advantage of rule-set classifiers is that they are often more accurate predictors than decision trees, since the rule set has an error rate of 0.5% on test cases. For very large datasets, however, generating rules with the rule set option may require much more computer time.

Naive Bayes Classifier

Naive Bayes is a probabilistic classifier, which means that for a document d, all classes c ∈ C the classifier returns the class c which has the posterior maximum probability given the document.

The Naive Bayes classifier is a simple probabilistic classifier that is based on the Bayes theorem with strong and naive assumptions of independence. This is one of the most basic text classification techniques with various applications in the detection of e-mail, personal message sorting, categorization of documents, detection of sexually explicit content, detection of languages and The detection of feelings. Despite the naive design and simplified assumptions that this technique uses, Naive Bayes works well in many complex real world problems.

Even though it is often surpassed by other techniques such as boosted trees, random forests, Max Entropy, Support Vector Machines etc., Naive Bayes classifier is very effective as it is less costly in computing (both CPU and memory) And it requires a small amount of training data. In addition, the training time with Naive Bayes is much lower as opposed to the alternative methods.

The Naive Bayes classifier is superior in terms of CPU and memory consumption as shown by Huang, J. (2003), and in many cases its performance is very similar to more complicated and slower techniques.

A naive bayes classifier[15] is a simple probabilistic model based on the Bayes rule with a strong hypothesis of independence. The Naïve Bayes model implies a simplified conditional independence hypothesis. This is given a class (positive or negative), the words are conditionally independent of each other. This assumption does not significantly affect the accuracy of the text classification, but makes the classification algorithms very fast applicable to the problem. In our case, the probability of maximum likelihood of a word belonging to a given class is given by the expression:

$$P(x|c) = \frac{countofproductxinclassc}{totalnumberofproductsinclassc}$$

Here, the xi s are the individual words of the post tweet. The classifier delivers the class with the maximum a posteriori probability. We also remove duplicate words from tweets, they do not add any additional information; This type of Naive Bayes algorithm is called Bernoulli Naïve Bayes. The inclusion of the presence of a word instead of the count has been found to improve performance marginally, when there are a large number of training examples

IV. KEY INDEX PARAMETERS FOR RESULT CLASSIFICATION

In prediction systems, most important is the final result obtained from the users. In fact, in some cases, users doesn't care much about the exact ordering of the list a set of few good recommendations is fine. Taking this fact into evaluation of recommender systems, we could apply classic information retrieval metrics to evaluate those engines: 1. Precision 2. Recall and 3. F1-Score. These metrics are widely used on information retrieving scenario and applied to domains such as search engines, which return some set of best results for a query out of many possible results [34][27]. For a search engine for example, it should not return irrelevant results in the top results, although it should be able to return as many relevant results as possible. Precision is the proportion of top results that are relevant, considering some definition of relevant for your problem domain. The Precision at 10 would be this proportion judged from the top 10 results. The Recall would measure the proportion of all relevant results included in the top results. In a formal way, we could consider documents as instances and the task it to return a set of relevant items given a search term. So the task would be assigning each item to one of two categories: relevant and not relevant. Recall is defined as the number of relevant items retrieved by a search divided by the total number of existing relevant items, while precision is defined as the number of relevant items retrieved by a search divided by the total number of items retrieved by the search

In information retrieval with binary classification, precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also called sensitivity) is the fraction of the relevant instances that are retrieved. Precision and recall are therefore based on understanding and measuring relevance.

In simple terms, high accuracy means that an algorithm returns significantly more relevant than irrelevant results, while a high recall means that an algorithm has yielded the most relevant results.
The most important category measurements for binary categories are:

Precision	Recall	F Measure
$P = TP/(TP + FP)$	$R = TP/(TP + FN)$	$tp + tn/tp + tn + fp + fn$

V. Conclusion

For E-commerce projects, and particularly in the E-commerce mobile market or portal, the ideal situation is to develop a hybrid technique with many active contributors that provides a rich and varied set of recommender system functions that meets all or most of the baseline development requirements. Short of finding this ideal solution, some minor customization to an already existing system may be the best approach to meet the specific development requirements. Various libraries have been released to support the development of recommender systems for some time, but it is only relatively recently that larger scale, open-source platforms have become readily available. In the context of such platforms, evaluation tools are important both to verify and validate baseline platform functionality, as well as to provide support for testing new techniques and approaches developed on top of the platform. Apache Mahout as an enabling platform for research and have faced both of these issues in employing it as part of work in collaborative filtering recommenders.

References

[1] Yueping Wu and Jianguo Zheng, “A Collaborative Filtering Recommendation Algorithm Based on Improved Similarity Measure Method”, Proc. of Progress in informatics and computing, 2010, pp.246- 249.

[2] Xiwei Wang, Erik von der Osten, Xuzi Zhou, Hui Lin and Jinze Liu, “Case Study of Recommendation Algorithms”, Proc. of International Conference on Computational and Information Sciences, 2011, pp.410 417.

[3] Xue, W., B. Xiao, et al. (2015). Intelligent mining on purchase information and recommendation system for e-commerce. 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).

[4] Zheng, Y., C. Liu, et al. (2016). Neural Autoregressive Collaborative Filtering for Implicit Feedback. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. Boston, MA, USA, ACM: 2- 6.

[5] Zhao, Q., Y. Zhang, et al. (2017). Multi-Product Utility Maximization for Economic Recommendation. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. Cambridge, United Kingdom, ACM: 435-443.

[6] Zheng, L., V. Noroozi, et al. (2017). Joint Deep Modeling of Users and Items Using Reviews for Recommendation. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. Cambridge, United Kingdom, ACM: 425-434.

[7] Najafabadi, M. K. and M. N. R. Mahrin (2016). "A systematic literature review on the state of research and practice of collaborative filtering technique and implicit feedback." Artificial intelligence review 45(2): 1- 35.

[8] Luo, L., B. Li, et al. (2016). Discovering Temporal Purchase Patterns with Different Responses to Promotions. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. Indianapolis, Indiana, USA, ACM: 2197-2202.

[9] Li, D., G. Zhao, et al. (2015). A Method of Purchase Prediction Based on User Behavior Log. 2015 IEEE International Conference on Data Mining Workshop (ICDMW).

[10] Wang, Y., V. Chattaraman, et al. (2015). "Predicting Purchase Decisions Based on Spatio-Temporal Functional MRI Features Using Machine Learning." IEEE Transactions on Autonomous Mental Development 7(3): 248-255.

[11] Wu, S., W. Ren, et al. (2016). Personal recommendation using deep recurrent neural networks in NetEase. 2016 IEEE 32nd International Conference on Data Engineering (ICDE).

[12] Yi, Z., D. Wang, et al. (2015). Purchase Behavior Prediction in MCommerce with an Optimized Sampling Methods. 2015 IEEE International Conference on Data Mining Workshop (ICDMW).

[13] Wu, C.-Y., C. V. Alvino, et al. (2016). Using Navigation to Improve Recommendations in Real-Time. Proceedings of the 10th ACM Conference on Recommender Systems. Boston, Massachusetts, USA, ACM: 341-348.

[14] Li, Q., M. Gu, et al. (2015). Multi-Classes Feature Engineering with Sliding Window for Purchase Prediction in Mobile Commerce. 2015 IEEE International Conference on Data Mining Workshop (ICDMW).

[15] Dong, H., L. Xie, et al. (2015). Research on statistics-based model for Ecommerce user purchase prediction. Computer Science & Education (ICCSE), 2015 10th International Conference on.

[16] Boroujerdi, E. G., S. Mehri, et al. (2014). A study on prediction of user's tendency toward purchases in websites based on behavior models. Information and Knowledge Technology (IKT), 2014 6th Conference on.

[17] Chadha, A. and P. Kaur (2015). Comparative analysis of recommendation system. Emerging Trends and Technologies in Libraries and Information Services (ETTLIS), 2015 4th International Symposium on.

[18] Rogers, S. K. (2016). Item-to-item Recommendations at Pinterest. Proceedings of the 10th ACM Conference on Recommender Systems. Boston, Massachusetts, USA, ACM: 393-393.

[19] Silva, E. d. S. d. (2017). New Probabilistic Models for Recommender Systems with Rich Contextual and Content Information. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. Cambridge, United Kingdom, ACM: 839-839.
[20] Son, L. H. (2016). "Dealing with the new user cold-start problem in recommender systems: A comparative review." Information Systems 58: 87-104.
[21] Pourgholamali, F. (2016). Mining Information for the Cold-Item Problem. Proceedings of the 10th ACM Conference on Recommender Systems. Boston, Massachusetts, USA, ACM: 451-454.
[22] Balakrishnan, R. and S. Kambhampati (2016). "Click efficiency: a unified optimal ranking for online Ads and documents." Journal of Intelligent Information Systems 46(3): 531-562.
[23] Jawaheer, G., Weller, P., & Kostkova, P. (2014). Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback. ACM.
[24] Lo, C., D. Frankowski, et al. (2016). Understanding Behaviors that Lead to Purchasing: A Case Study of Pinterest. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA, ACM: 531-540.