IEICE TRANS. INF. & SYST., VOL.E99–D, NO.1 JANUARY 2016

LETTER

Purchase Behavior Prediction in E-Commerce with Factorization Machines

Chen CHEN[†a)], ***Member***, **Chunyan HOU**[††b)], **Jiakun XIAO**[†], ***and*** **Xiaojie YUAN**[†], ***Nonmembers***

SUMMARY Purchase behavior prediction is one of the most important issues for the precision marketing of e-commerce companies. This Letter presents our solution to the purchase behavior prediction problem in E-commerce, specifically the task of Big Data Contest of China Computer Federation in 2014. The goal of this task is to predict which users will have the purchase behavior based on users' historical data. The traditional methods of recommendation encounter two crucial problems in this scenario. First, this task just predicts which users will have the purchase behavior, rather than which items should be recommended to which users. Second, the large-scale dataset poses a big challenge for building the empirical model. Feature engineering and Factorization Model shed some light on these problems. We propose to use Factorization Machines model based on the multiple classes and high dimensions of feature engineering. Experimental results on a real-world dataset demonstrate the advantages of our proposed method.

key words: *purchase behavior, prediction, e-commerce, Factorization Machines*

1. Introduction

E-commerce, also known as online shopping, is becoming more popular as more consumers look to the Internet for purchasing products. There is an increasing percentage of e-commerce in the retail sales trade. E-commerce companies try their best to meet the various needs of customers and promote their sales. Marketing is emphasized for e-commerce companies, such as Amazon and eBay, to guide online consumers to an e-commerce website and persuade them to buy the products or services online. In addition, e-commerce makes it easy for these companies to track the behaviors of customers. Thus, analyzing customers historical behaviors and predicting their purchase behaviors have become critical issues in helping them to find products they will like and purchase.

The goal of the purchase behavior prediction in e-commerce[*], as one task of Big Data Contest[**] of China Computer Federation in 2014, is to encourage the development of empirical models to predict the purchase behaviors of consumers. The prediction would help in several practical scenarios, including: 1) improve personalized recommendation system by predicting customers who will have the purchase behavior in an e-commerce website; 2) provide e-commerce companies with suggestions for the effective advertisement.

Most studies have analyzed the behaviors of customers to predict their product preferences. Generally, we can divide behaviors into two types: explicit and implicit. An example of explicit behaviors is that a customer tells us which products he like or dislike while implicit behaviors cannot demonstrate users' product preference directly. These implicit behaviors include user purchase patterns, web page visits and web browsing paths [1]. In addition, Lee et al. propose that customers of online stores go through four main shopping steps: product impression, click-through patterns, basket placement and purchase [2]. The product impression means that the advertisement of a product in a media is viewed. The click-through indicates that the advertisement is clicked and the web page of the product is seen. Other researchers use customers' behavioral data, which includes clicking, adding to cart and purchasing, for recommendation system [3]–[5]. Although our study also involves implicit behaviors, the task differs from those studies because the prediction of customers who will purchase is the aim regardless of items or products. The most similar work is the prediction of repurchase rates, which is defined as the probability of each customer purchase at least once one product with respect to a specific category in the electronic shop [6]. In contrast to the prediction of repurchase rate, our work focuses on the short-term purchase prediction based on implicit behaviors because the short-term prediction shed light on the effectiveness of product impression and clicking behaviors to purchase behaviors timely.

This letter presents our empirical model to predict the short-term purchase behavior of consumers in E-commerce. Unlike existing studies on recommendation system, this task just predicts which users will have the purchase behavior, rather than which items should be recommended to which users. Thus, some classical models of recommendation system are not proper for this task. On the other hand, the large-scale dataset is a big challenge for learning the empirical model.

We define the problem as follows. Let U be the set of users and V be the set of advertisements. An event is observed when a user views a web page with advertisements

Manuscript received May 15, 2015.
Manuscript revised August 28, 2015.
Manuscript publicized October 1, 2015.

[†]The authors are with the College of Computer and Control Engineering, Nankai University, Tianjin, China.
[††]The author is with the School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China.
a) E-mail: nkchenchen@nankai.edu.cn
b) E-mail: houchunyan@tjut.edu.cn (Corresponding author)
DOI: 10.1587/transinf.2015EDL8116

[*]http://www.semidata.com/BDC2014.html
[**]http://bigdatacontest.ccf.org.cn/problems.html

or clicks the URL of an advertisement. For example, we observe that a user clicks the URL of car advertisement in context C, and C includes Time, IP address and so on. Given all observed events, the goal of the task is to predict which users will have the purchase behavior.

2. Method

2.1 Dataset

The contest data consists of training, validation and testing datasets. The training dataset is used to build the empirical model and results are asked to submit based on the testing dataset. In terms of one brand in an E-commerce site, datasets include product impressions and users' clicking behaviors for the part of users during one period of time, as well as their transaction records. The Table 1 shows information of fields in the dataset. Note that each advertisement is unique and has its own advertisement ID. The monitor indicates the media of advertising such as the portal or video webpage. For the training dataset, we have known which users have the purchase behavior in transaction records, which include user ID and purchasing time. These users are called transformed users in this task.

The statistics of training, validation and testing datasets are shown in Table 2. Note that behaviors include impression and clicking behaviors. There are a lot of users in the training dataset, and we find that most of them do not have click behavior and users without click behavior usually do not purchase any items. Thus, we extract data of 353,552 users with click behaviors in the training dataset to build the empirical model. For all these models, we set optimal parameters by the validation dataset. In addition, we have two findings. First, the number of behaviors is large and the scale-out dataset is a big challenge to build model. Second, compared with total users, the number of transformed users is very small and the extreme imbalance of the dataset makes it difficult to train the effective classification model.

Table 1 Fields of datasets

Field	Description
User	Unique user in e-commerce
User stability	Whether stable user
Advertisement	Unique advertisement in e-commerce
Monitor	Media of advertising. eg. portal, video webpage
Browser type	Browser. eg. IE, Chrome and etc
OS type	Operation System. eg. WinXP, Win8 and etc
Language	Language of the browser
IP address	IP address of the user
Time	Time of behavior
Behavior type	Impression and click

Table 2 Statistics of datasets

	Training	Validation	Testing
#User	1,930,999	633,022	633,264
#Transformed User	1389	465	477
#Advertisement	1,629	1,411	1,441
#Monitor	22,196	19,641	19,581
#Browser	226	204	206
#Operation System	17	17	17
#Language	49	30	33
#IP	7,960,034	2,804,239	2,810,482
#Day	30	30	30
#Behavior	162,755,706	53,657,677	52,787,677

2.2 Feature Engineering

Our feature engineering is one of the relevant contributions of this letter and an important part of our model. There are totally 16 features in Table 3. We compute all feature values on the large scale dataset in a Hadoop cluster to accelerate data processing.

We give short names for features in Feature column and explain these features in Description column. The behavior includes the impression and the click. The more behaviors a user has, the more possibly the user have the purchase behavior. The user who views more advertisements is more likely to buy items. Specifically, IP-Change feature describes the transition frequency of IP addresses used by users for the electronic shopping. If some users visit items at work and purchase items at home, this feature can exhibit users preference. Active users can be distinguished by Click-Rate, which is the number of clicks divided by the number of behaviors for each user. We regard a day as the day of abnormal click when the number of click this day is more than twice of the average number of click per day. The abnormal behavior is similar. Note these features are in terms of each user, that is, each user has 16 features.

We define the feature set A which includes all features in Table 3. We investigate the contributions of these features by means of ranking users by single feature value in the descent order and computing the F1-score@2000. We observe that Adv, Adv-CaB, Click-Rate, Days-AC features are more effective than others. Inspired by this observation, we decide to design a lot of extended features based on advertisements, clicks and days. Besides these features of set A, we design four classes of features by combining advertisements, behaviors and days, which include viewed advertisements, clicked advertisements, the day of viewing and

Table 3 Features

Feature	Description
Behavior	Total #behaviors
Impression	Total #impression
Click	Total #clicks
Adv	#(unique advertise)
Monitor	#(unique monitors)
IP	#(unique IP addresses)
Adv-CaB	#(advertises clicked after viewed)
IP-Change	#(changes of IP address)
Adv-Click	#(advertise clicked)
Click-Rate	#clicks divided by #behaviors
Days-AB	#(days of abnormal behavior)
Days-AC	#(days of abnormal click)
Max-Adv-Click-Rate	Max click rate of advertise
Max-Behavior	Max #behaviors of day
Max-Click	Max #clicks of day
Max-Click-Adv	Max #clicks of advertise

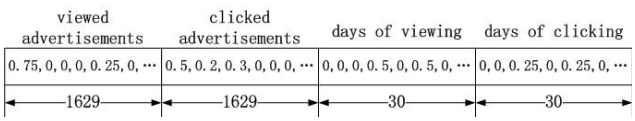

Fig. 1 Illustration of the vector representation of feature set B

day of clicking. We use high-dimension vector to represent each class of feature.

As shown in Fig. 1, the vector of viewed advertisements indicates that which advertisements are viewed and how many times they are viewed by the user. Note that each element of the vector is associated with a unique advertisement. The feature values are normalized by dividing the total number of views. For illustration in Fig. 1, the first element is 0.75 and the fifth element is 0.25 in the feature vector of viewed advertisements. It means that the first and fifth advertisements are viewed and the proportion of views is 3 : 1. This representation of the feature vector is able to exhibit the preference of the user's views exactly. Clicked advertisements are in the same way. The feature vector of days of viewing shows that on which days the user views the advertisements. Every element of the vector corresponds to the day of a month. For example in Fig. 1, the fourth and sixth elements are 0.5 in the feature vector of day of viewing, which reveals the user views advertisements on the fourth and sixth days of the month. This feature vector can exploit the user's preference to the date. The day of clicking is similar.

Because there are 1629 advertised and 30 days in the training dataset, the dimension of viewed or clicked advertisements is 1629 and day of viewing or clicking is 30. The total dimension of the feature vector are $(1629 + 30 + 1629 + 30) = 3318$. Four classes of features are included in feature set B and the number of features in set B is 3318. The following section will compare the performance of two set of features.

2.3 Model

Naive Bayes, Logistic Regression, Support Vector Machine are chose for this task. Naive Bayes, Logistic Regression are proper for large scale datasets. Support Vector Machine is one of typical classification model. We use support vector classfication provided by LIBSVM[†]. Recently, Factorization Machines[††], presented by Rendle [7], are used in some recommendation application and have shown excellent prediction capabilities. Factorization Machine is a state-of-the-art framework for the factorization model with a variety of features.

It is able to model all nested interactions up to order d between the p features in x using factorized interaction parameters. The model of order $d = 2$ is defined as:

[†]http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[††]http://libfm.org/

Table 4 Experimental results

Feature Set	Model	P@2000	R@2000	F1-Score
A	NB	0.019	0.0797	0.0307
A	LR	0	0	0
A	C-SVM	0	0	0
A	FM	0.0215	0.0901	0.0347
B	NB	0.0115	0.0482	0.0186
B	FM	0.0480	0.2013	0.0775
A+B	NB	0.0115	0.0482	0.0186
A+B	FM	0.0510	0.2100	0.0821

$$\hat{y}(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} \left\langle v_i, v_j \right\rangle x_i x_j \tag{1}$$

where k is the dimensionality of the factorization and the model parameter is:

$$w_0 \in \mathbb{R}, \quad w \in \mathbb{R}^n, \quad V \in \mathbb{R}^{n \times k} \tag{2}$$

In addition, $\left\langle v_i, v_j \right\rangle$ is the dot product of two k-dimension vectors:

$$\left\langle v_i, v_j \right\rangle = \sum_{f=1}^{k} v_{i,f} v_{j,f} \tag{3}$$

As shown in Eq. (1), for this task y denotes the probability of a user's purchase behavior after viewing or clicking the advertisements. The first part (i.e. w_0) means the bias of the user. The larger w_0 is, the more likely the user is to purchase products. The second part models the interaction of each feature x_i with y. The third part contains all pairwise interactions (i.e. $x_i x_j$) because the combination of two features can affect the purchase behavior. For example, a user decides to buy after he views an advertisement, and then clicks this advertisement on the same day.

Markov Chain Monte Carlo (MCMC) inference is used for learning model because it is the easy to process large scale dataset.

2.4 Result

As shown in Table 2, there are 477 transformed users and 633,264 users in the testing dataset. We use Pecision@2000(P@2000), Rcall@2000(R@2000) and F1-Score as evaluation metrics to measure the prediction quality. F1-Score is calculated by P@2000 and R@2000. Table 4 presents the results of all methods on the large scale dataset. In the Contest, our score is in the top 10 out of about 900 teams. The result can validate the significant performance of our proposed factorization machines model with the feature set A and B.

We compare the following methods: Naive Bayes (NB), Logistic Regression (LR), Classification SVM (C-SVM) and Factorization Machines (FM). For this problem, these methods can output the probability of purchasing behavior for each user. We get top 2000 users by the probability in the descent order.

With the features of set A, LR and C-SVM predict that all users do not purchase and get zero precision and recall.

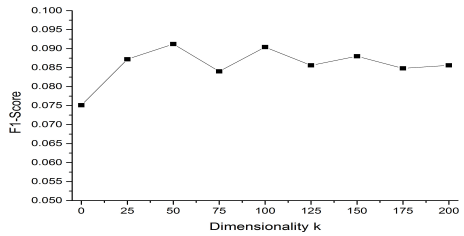

Fig. 2 F1-score vs. dimensionality k

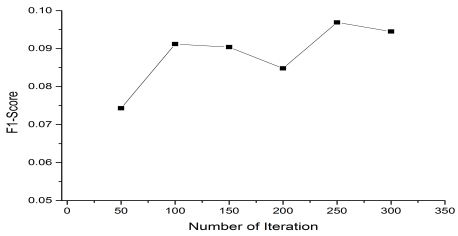

Fig. 3 F1-score vs. number of iteration

The most possible reason is that the number of transformed users is very small, that is, the users with the purchase behavior account for a few portion of all users. LR and C-SVM are not proper in this case. NB and FM work effectively to some extent, and FM performs better than NB moderately.

With the high dimension features of set B and set $(A+B)$, LR and C-SVM cannot finish in reasonable time due to the scale of dataset. So we only report the performance of NB and FM in Table 4. FM achieves better performance than NB in terms of F1-Score. FM is stable when the number of features increases, while NB has the worse result with set B features than set A features.

2.5 Sensitivity of Dimensionality

The factorization dimensionality k is an important parameter. When applying FM to purchase behavior prediction, we would like to know the performance of feature interactions. Thus, we train FM model with different k.

As shown in Fig. 1, when the dimensionality changes, the F1-score is almost similar. We observe that FM gets the best performance with $k = 50$. In addition, the improvement over FM with $k = 0$ is significant, which shows that 2-degree FM is more effective in purchase behavior prediction because the interactions between features are taken into account. On the other hand, when k becomes large, the performance of 2-degree FM is stable while it is better than FM with $k = 0$. Therefore, it is important to select an appropriate k. The factorization parameter also makes 2-degree FM more adaptive by adjusting the number of factors k for specific applications.

2.6 Sensitivity of Iteration Number

Another important parameter is the number of iteration for Markov Chain Monte Carlo (MCMC) inference. We would like to know how many iteration is enough for learning model so as to provide good predictive performance. We try to set the number of iteration from 50 to 300 with the step size 50. As shown in Fig. 2, we observe that the FM model can achieve optimal quality when the number of iteration is set to 250. In addition, the improvements over the model with 50 iterations are significant, which also reveals that no less than 100 iteration is necessary to learn an effective model.

3. Conclusions

This letter presents our solution to the task of purchase behavior prediction in Big Data Contest of China Computer Federation in 2014. The extreme scale and imbalance of the dataset are two big challenges for building empirical model. The traditional models do not work in this scenario. Feature engineering is an important part of our model and we learn Factorization Machines model on the large-scale dataset. Our experimental results demonstrate the effectiveness of our model. Furthermore, we investigate the sensitivity of the dimensionality and the number of iterations for building an effective predictor.

Future work includes putting this model into service to provide useful suggestions for promotions. In addition, it may be beneficial to use this study as the basis of a recommendation system so as to increase the ease of the customers' shopping enjoyment.

Acknowledgments

Research was partially supported by National Natural Science Foundation of China (No.61170184, 61402242, 61402243, 61402333), Research Foundation of Ministry of Education and China Mobile Communications Corporation (No.MCM20130381), Tianjin Municipal Science and Technology Commission (No.13ZCZDGX02200, 13ZCZDGX01098, 13JCQNJC00100, 15JCQNJC00400), National 863 Project of China (No.2013AA013204), Ph.D. Programs Foundation (No.20120031120038) and Fundamental Research Funds for the Central Universities.

References

[1] T. Lee, Y. Park, and Y. Park, "A time-based approach to effective recommender systems using implicit feedback," Expert Systems with Applications, vol.34, no.4, pp.3055–3062, 2008.

[2] J. Lee, et al., Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising, Data Mining and Knowledge Discovery, vol.5, no.1-2, pp.59–84, 2001.

[3] Q. Liu, E. Chen, H. Xiong, C.H.Q. Ding, and J. Chen, "Enhancing collaborative filtering by user interest expansion via personalized ranking," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.42, no.1, pp.218–233, 2012.

[4] Y.-J. Park and K.-N. Chang, "Individual and group behavior-based customer profile model for personalized product recommendation," Expert Systems with Applications, vol.36, no.2, pp.1932–1939, 2009.

[5] T.C.-K. Huang, "Mining the change of customer behavior in fuzzy

time-interval sequential patterns," Applied Soft Computing, vol.12, no.3, pp.1068–1086, 2012.

[6] Y.-H. Wang, R.-D. Chiang, and H.-C. Chu, "Combing Customer Profiles for Members' Repurchase Rate Predictions," Journal of Software, vol.8, no.6, pp.1316–1326, 2013.

[7] S. Rendle, "Factorization machines with libfm," ACM Transactions on Intelligent Systems and Technology, vol.3, no.3, pp.1–22, 2012.