Predicting Purchase Behavior of E-commerce Customer, One-stage or Two-stage?

Chen CHEN¹, Chun-yan HOU²*, Jia-kun XIAO¹ and Xiao-jie YUAN³

¹Nankai University, Tianjin, China
²Tianjin University of Technology, Tianjin, China
³*Corresponding author

Keywords: Recommender system, Purchase Prediction, Behavior Analysis

Abstract. The development of Internet has led to huge economic benefits and challenges for e-commerce companies, which develop online shopping stores to serve online customers, as well as collect valuable data of customers' behavior. Purchase behavior prediction is one of the most important issues to promote both companies' sales and consumers' satisfaction. This paper studies two modeling strategies of purchase behavior prediction in the engineering practice: one-stage and two-stage. The appropriate modeling strategy is essential to implement an effective and efficient predictive model. The pros and cons of one-stage and two-stage modeling is analyzed and discussed. We also conduct experiments on real-world and large-scale dataset and find that the performance of two-stage modeling is usually better than one-stage. The reason is that two-stage modeling can make better use of the feature engineering for each stage and reduce the requirement for machine learning algorithms. The purchase behavior can be better predicted by the machine learning algorithm which is ensemble and able to use high-order interactions among features.

Introduction

As online shopping becomes popular, behavior analysis of e-commerce customers has become an increasingly important business tool for promoting sales. Many e-commerce companies, such as Amazon¹ in USA and Taobao² in China, are looking for all possible approaches to detect customers' intent and predict their purchase behaviors. Both the consumers and online businesses benefit from this kind of prediction technique. For example, a user, who is not registered and a temporary visitor, can be attracted by recommending him the discounted item if the user's preference for this item is determined by his short-term browsing information. The ability to predict what a customer will purchase is also valuable for e-commerce companies. Exact predictions for the next products to be purchased enable companies to implement a product recommendation system, determine the positions of products in the result of a customer's search query, optimize the collection of products to be displayed or suggest products to complement the contents of a customer's shopping basket. However, the prediction becomes difficult when there is only the http session of a user which consists of a list of clicks in a short time.

The prediction with the short-term history becomes an area of growing research and commercial interest in recent years. On one hand, session-based prediction has been studied firstly in the music recommendation. Session-based collaborative filtering [3], session-based hierarchical graphical model [9] are proposed for the music recommendation. On the other hand, there are some researches about purchase behavior prediction, which are based on the long-term and stable users' information, such as demographics [2], media advertisements [5] and the social media [8]. However, few work has been done to research purchase behavior prediction based on e-commerce sessions. The main reason is the lack of real-world data of customers' purchase behaviors. Fortunately, RecSys 2015 Challenge provides a large number of sessions from an online e-commerce retailer [1]. A history of user's click

¹ http://www.amazon.com/
² http://www.taobao.com
behavior during a browsing session at a website of online retailer is given, and the goal is to predict the items the user will purchase at the end of this session.

The requirement of the challenge is naturally to answer two following questions: 1) which sessions are going to contain buy events? and 2) which items are bought in that session? Each question can correspond to one task. Thus, there are two modeling strategies for this challenge: one-stage and two-stage. If two tasks are completed one after another, it is called two-stage modeling. This strategy seems to be complicated in the sense that the second task can achieve the same result. If the second task decides not to purchase any items, it will result in the same effect as the first task determines that there is not buy events in this session. If the second task is just considered, it is named one-stage modeling. In theory, it seems quite natural to divide the prediction into two stages. However, the modeling strategy is a really important issue for implementing the predictive system in the engineering practice. From the perspective of engineering, both the effectiveness and efficiency are taken into account because of the large scale training data. The modeling strategy is closely associated with the effectiveness and efficiency.

In this paper, we focus on comparing pros and cons of the one-stage and two-stage modeling strategies in the engineering practice. In the RecSys 2015 Challenge, both strategies are used by participants to achieve top-ranking performance [4,6,7]. The appropriate modeling strategy is essential to implement an effective and efficient predictive model, which is able to be easily optimized by the specified metric.

The rest of the paper is organized as follows. In Section 2, we describe two modeling strategies and compare them. In Section 3, we conduct experiments to study two modeling strategies for the purchase behavior prediction and analyze the results in details. We conclude our paper in Section 4.

Modeling Strategy

Problem Definition

The dataset is divided into train and test dataset. Train and test dataset consist of sets of sessions \( S_{\text{train}} \) and \( S_{\text{test}} \) respectively. Each session \( s \) is represented as a click stream

\[
c(s) = (c_1(s), c_2(s), \ldots, c_{n(s)}(s))
\]

where \( n(s) \) is the number of clicks in session \( s \), \( c_j(s) \) denotes the \( j \)-th clicked item in session \( s \), \( t_j(s) \) is the time when item \( c_j(s) \) is clicked, \( y_j(s) \) is one when item \( c_j(s) \) is purchased at least once, and zero otherwise. A session \( s \) has the label \( y(s) \) defined as

\[
y(s) = \begin{cases} 
1 & \exists j : y_j(s) = 1 \\
0 & \forall j : y_j(s) = 0 
\end{cases}
\]

If \( y(s) = 1 \), session \( s \) is a buy session. We are given sets of purchased items for session \( s \in S_{\text{train}} \), and are required to predict these sets for session \( s \in S_{\text{test}} \). In two-stage modeling, \( y(s) \) is predicted, and then \( y_j(s) \) is predicted. In one-stage modeling, \( y_j(s) \) is directly predicted, and then \( y(s) \) is inferred by \( y_j(s) \).

One-stage Modeling

One-stage modeling focuses on which items are bought in a session. The one-stage modeling is illustrated in Figure 1. To begin with, we extract two groups of features from train dataset: session features and item features. In the same way, two groups of features are extracted from validation dataset and test dataset. The labels of train samples can be obtained by buy dataset and we train a machine learning model (i.e. item model) which is used to predict the purchase. Then we use the item model to predict samples of the validation dataset and select the best threshold \( \beta \) which can achieve
the best evaluation score on the validation dataset. Finally, the item model and the threshold are applied to test dataset. In the sense, if none of items is bought, it will in effect be the same as if it is not a buy session.

Two-stage modeling uses the part of the entire train dataset, which are remained after the first stage, to train the item model. Unlike the two-stage modeling, one-stage modeling can make full use of the entire train dataset. Furthermore, a sample of train dataset in one-stage modeling is associated with a click on an item in a session. For example, a session consists of 10 clicks on 5 unique items and there are 10 training samples based on this session. There are about 9 million sessions and about 33 million clicks. Thus, about 33 million samples can be used to train the item model in one-stage modeling. The number of samples in one-stage modeling is much larger than that of two-stage. Machine learning algorithm is required to be scalable and more time and memory are spent to train a predictive model. In addition, there are fewer parameters (e.g. threshold $\beta$) for one-stage modeling to be tuned than two-stage.

Two-stage Modeling

As shown in Figure 2, we firstly extract session features from train dataset and their labels can be obtained by buy dataset. We train a session model to estimate purchase. By the same means, session features are extracted from validation dataset. After selecting an appropriate threshold $\alpha$ by validation dataset, we use $\alpha$ to remove sessions from test dataset that do not pass this threshold. Then, session features and item features are extracted from train dataset for items which are bought in buy dataset. We build an item model based to two kinds of features and predict the purchase for kept sessions of test dataset. The two-stage modeling firstly determines whether a session is going to buy an item or not, and if so, which items will be purchased.

The advantage of two-stage modeling is that different problems are emphasized in each stage. In the first stage, the training samples are based on sessions, rather than items, and the number of training samples is fewer than that of one-stage. Because sessions which are predicted to buy at least an item are kept in train dataset, the number of samples in train dataset used in the second stage decreases. For the large-scale dataset, two-stage modeling makes the training more efficiently. However, it is possible for the first stage to remove buy sessions and the samples based on those sessions are also ignored by the item model in the two-stage modeling. Moreover, two stages are processed separately and more parameters need to be tuned for each of two stages.

Experiment

Dataset

Train and validation dataset are generated by applying 75% and 25% split to the original train dataset. In that case, the validation dataset has the same size of the test dataset. Our aim is to compare two different modeling strategies and reveal that the modeling strategy is an important issue in the engineering practice. We randomly select some samples from train, validation, and test dataset respectively. There are 762,681 sessions, 3,000,009 clicks and 28,767 items in training dataset,
255,116 sessions, 1,000,004 clicks and 22,376 items in the validation dataset while 254,009 sessions, 1,000,000 clicks and 22,424 items in the testing dataset.

Experiment Design and Setup

For all experiments, we use train dataset to learn a model and tune all parameters by validation dataset. We compare the performance of Logistic Regression (LR), Factorization Machines (FM) and Gradient Boosting decision tree (GB) in terms of one-stage and two-stage modeling.

We have two groups of features. One group describes a session and the other illustrates an item in a session. The session model just uses session features and the item model uses both two groups of features. The features used in the experiments are listed in Table 1. We use explicit features which are extracted directly from the raw dataset. These features are independent and simple to investigate the effect of interactions among features for purchase behavior prediction.

For logistic regression, we use LibLinear\(^3\) which is a library for the large linear classification. The standard \(L_2\) regularization is used in the experiments. For factorization machines, we use LibFM\(^4\) which is an implementation of factorization machines. Markov Chain Monte Carlo is used to learn the model. For gradient boosting decision tree, we use the XGBoost\(^5\) library. We concentrate on the binary classification with the logistic loss function, and use tree boosters and \(L_2\) regularization term on weights.

Result Analysis

Table 2 reports the results of machine learning algorithms with different modeling strategies on test dataset. The \textbf{stage} column indicates the modeling strategy which is one-stage (one) or two-stage (two). While \textbf{S-P} and \textbf{S-R} columns represent the precision and recall of sessions, \textbf{I-P} and \textbf{I-R} columns are the average item precision and recall per session respectively. \textbf{S-Score}, \textbf{I-Score} and \textbf{Score} columns are the session score, the item score and score used in the challenge. The paired t-test is used for the statistical significance with a significance level \(p=0.05\). The score with the star means that the improvements over one-stage modeling are statistically significant.

As shown in Table 2, the modeling strategy has the impact on the evaluation metric when the predictive system is implemented and it is necessary to take it into account in the engineering practice. The performance of two-stage modeling is usually better than one-stage. LR and FM achieve

---

\(^3\) https://www.csie.ntu.edu.tw/~cjlin/liblinear/
\(^4\) http://www.libfm.org/
\(^5\) https://github.com/dmlc/xgboost/
significantly more scores in two-stage than one-stage. Although GB gets the higher score with one-stage modeling, the improvements are not statistically significant. Let us analyze the results in detail. First, for the session precision and recall, session model is usually better in two-stage modeling than one-stage. The reason may be that two-stage modeling can make better use of feature engineering to improve the session prediction. The only exception is that the session precision is higher in one-stage modeling for gradient boosting model at the cost of the session recall. Second, from the item precision and recall points of view, the performance in two-stage modeling outperforms one-stage for LR and FM. LR and FM benefit from the dataset reduced by the former part of two-stage. The performance of GB in two-stage modeling is worse than one-stage, but the difference is not statistically significant. The possible reason is that one-stage modeling is able to use the entire train dataset and the first procedure of two-stage removes some sessions from datasets which can be used by GB model. Third, the evaluation metric tends to be high when the session recall is high and the session precision is low, which can be observed from minus session scores. Both the item precision and item recall are important for the evaluation metric and it is better to keep similar values between the item precision and item recall.

Table 1. Features Used in the Experiments.

<table>
<thead>
<tr>
<th>Group</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session</td>
<td>click</td>
<td>#(clicks in the session)</td>
</tr>
<tr>
<td></td>
<td>time span</td>
<td>time span of the session</td>
</tr>
<tr>
<td></td>
<td>item</td>
<td>#(unique items clicked)</td>
</tr>
<tr>
<td></td>
<td>category</td>
<td>#(unique categories clicked)</td>
</tr>
<tr>
<td></td>
<td>month</td>
<td>month when the last item in the session is clicked</td>
</tr>
<tr>
<td></td>
<td>day of week</td>
<td>day of week when the last item in the session is clicked</td>
</tr>
<tr>
<td></td>
<td>Hour</td>
<td>hour when the last item in the session is clicked</td>
</tr>
<tr>
<td>Item</td>
<td>item id</td>
<td>the unique identifier of the item</td>
</tr>
<tr>
<td></td>
<td>category id</td>
<td>The unique identifier of the category</td>
</tr>
<tr>
<td></td>
<td>month</td>
<td>month when the item is clicked</td>
</tr>
<tr>
<td></td>
<td>day of week</td>
<td>day of week when the item is clicked</td>
</tr>
<tr>
<td></td>
<td>hour</td>
<td>hour when the item is clicked</td>
</tr>
<tr>
<td></td>
<td>interval</td>
<td>interval between the click of item and next item</td>
</tr>
<tr>
<td></td>
<td>position</td>
<td>position of item clicked in the session</td>
</tr>
<tr>
<td></td>
<td>item click</td>
<td>#(the item’s clicks in the session)</td>
</tr>
<tr>
<td></td>
<td>category click</td>
<td>#(clicks in the category that the item belongs to)</td>
</tr>
<tr>
<td></td>
<td>category item</td>
<td>#(items in the category that the item belongs to)</td>
</tr>
</tbody>
</table>

From the perspective of machine learning algorithms, FM is better than LR because FM can model pairwise interactions among features while LR cannot estimate the interactions. However, LR and FM suffer from the impact of the large-scale and imbalanced train dataset. GB outperforms both LR and FM. The machine learning algorithm, which can model high-order interactions among features, is effective for the large-scale dataset. The ensemble learning can alleviate the impact of imbalanced data and improve the prediction performance. Thus, GB usually can make full use of large-scale dataset and produces the competitive and robust model even though the train dataset is imbalanced. The top teams on the public leaderboard usually use two-stage modeling [6,7]. The possible reason is that they can design effective features in the different stages and refine train dataset through the first
stage. The scores of those teams reveal that complicated feature engineering can exhibit high-order interactions among features and help two-stage modeling to achieve the comparable score.

Conclusions

In this paper, we study the purchase behavior prediction of e-commerce customers in the engineering practice and focus on two modeling strategies: one-stage and two-stage. We conduct experiments with different machine learning algorithms and use explicit features to predict the purchase behaviors. We demonstrate that the performance of two-stage modeling is usually better than one-stage. The possible reason is that two-stage modeling can make better use of feature engineering for each stage and reduce the requirement for the machine learning algorithms. The ensemble learning algorithm which also models high-order interactions among features is suitable for purchase behavior prediction.

To the best of our knowledge, our study is the first attempt to analyze the modeling strategies of implementing the system in the engineering practice for purchase behavior prediction. We hope that it will help and inspire e-commerce companies to develop better recommendation systems and increase customers' engagement and satisfaction.

Acknowledgements

This research was partially supported by National Natural Science Foundation of China (No. 61402333, 61402242 and 61402243), National High Technology Research and Development Program 863 (No. 2015AA015401), the Research Foundation of Ministry of Education and China Mobile Communications Corporation (No. MCM20150507)

References


