

Purchase Prediction via Machine Learning in Mobile Commerce

Chao Lv, Yansong Feng* and Dongyan Zhao

Institute of Computer Science and Technology
Peking University, Beijing 100871, China
{lvchao, fengyansong, zhaodongyan}@pku.edu.cn

Abstract. In this paper, we propose a machine learning approach to solve the purchase prediction task launched by the Alibaba Group. In detail, we treat this task as a binary classification problem and explore five kinds of features to learn potential model of the influence of historical behaviors. These features include user quality, item quality, category quality, user-item interaction and user-category interaction. Due to the nature of mobile platform, time factor and spacial factor are considered specially. Our approach ranks the 26th place among 7186 teams in this task.

Keywords: Purchase Prediction, Machine Learning

1 Introduction

Recently, the Alibaba Group launched a purchase prediction task known as Ali Mobile Recommendation Algorithm ¹. This purchase prediction task provides the historical behaviors data of users in the mobile platform during a period of one month to help predict purchase behaviors will happen in the following one day. The historical behaviors include click, collect, add-to-cart and payment. Conventional methods in recommender system[6], such as collaborative filtering and matrix factorization, don't obtain a good performance in this task.

In this paper, we propose a machine learning approach to solve this purchase prediction task, instead of CF-based methods. This task is treated as a binary classification problem, and five kinds of features are explored from different aspects to learn potential model of the historical browsing behaviors, including user quality, item quality, category quality, user-item interaction and user-category interaction. Those features could reflect the willingness of users to buy items. In particular, we concentrate on the time and spacial factor. The time factor is incorporated into the feature families and features are extracted in different time dimension. The spacial factor is employed in the filtering module. For any purchase behaviors we predict, if the location of item is far away from the user, we will remove it from our prediction results.

*Corresponding author.

¹ <http://tianchi.aliyun.com>

2 Related Work

The most prominent technique in recommender system is Collaborative Filtering (CF) [8]. The basic insight for this technique is a sort of continuity in the realm of taste. If users Alice and Bob have the same utility for items 1 through k , then the chances are good that they will have the same utility for item $k + 1$. Usually, these utilities are based on ratings that users have applied for items with which they are already familiar. CF is roughly classified into two categories, i.e. memory-based approaches [5][9] and model-based approaches [1][3].

The Netflix million-dollar challenge boosted interest in CF and yielded the publication of a number of new methods. Several matrix factorization techniques have been successfully applied to CF, including Singular Value Decomposition (SVD) [7] and Non-negative Matrix Factorization (NMF) [4]. A joint non-negative matrix factorization method proposed in [2] tries to solve the purchase prediction task launched by the Alibaba Group in 2014. The goal of that task in 2014 is to predict purchase behaviors in the following one month based on historical behaviors data in a period of four months.

3 Problem Definition

Notations: U stands for the set of users, I stands for the whole set of items, P stands for the subset of items, $P \subseteq I$, D stands for the user behaviors data set in all the set of all items. Our objective is to develop a recommendation model for users in U on the business domain P using the data D . In detail, our goal is to predict purchase behaviors over P in the following one day based on the behaviors data during one month in D .

4 Method

We treat the target problem as a binary classification problem, i.e. any (*user*, *item*) pairs will be divided into two classes: “buy” and “not buy”. The framework of which is showed in Figure 1.

First, we would like to learn a model from the behaviors data over the whole set of items in the **training module**, which can reflect why users will buy items in the following one day and how their historical behaviors influence their future purchase behaviors. In detail, if *user* buy *item* in the following one day, this (*user*, *item*) pair will be labeled as a positive instance while other pairs that doesn’t be bought are going to be labeled as a negative instance. In addition, this trained model is applied to the behaviors data over the subset of items in the **prediction module** and positive instances in the prediction results will be seen as purchase behaviors will happen in the following one day. Then, we take spacial factor into consideration and remove pairs with too long distance in those positive instances via the **filtering module**. In the last, the filtered predicted purchase behaviors are compared with the real purchase behaviors to evaluate the performance of our approach in the **evaluation module**.

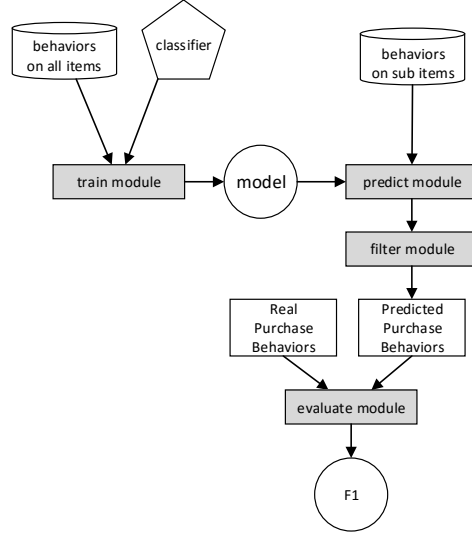


Fig. 1. The Machine Learning Diagram for Purchase Prediction

4.1 Training Module and Prediction Module

Training set is a basic component in the training module just like test set in the prediction module, but there is little difference between the generation of them. Because we can't use the future information, i.e. we don't know purchase behaviors on the whole set of items in the following one day, we split behaviors data in the last day of the month and use them to label $(user, item)$ pairs that appear in the remainder of the month. This process is illustrated in Figure 2.

4.2 Feature Project

Feature project is an important component in our machine learning approach and we will discuss feature families detailly in this section.

For a certain $(user, item)$ pair, the $item$ belongs to a $category$, we consider the following five feature families, i.e. user quality, item quality, category quality, user-item interaction and user-category interaction.

User Quality estimates the purchasing power and vitality of users. In the mobile commerce, some users are active and have strong purchase desire while others are inactive and not willing to buy items frequently.

- *Last Login Day* represents the last login day of a user.
- *Conversion Ratio* represents the ratio of purchase behaviors of a user in his total behaviors.

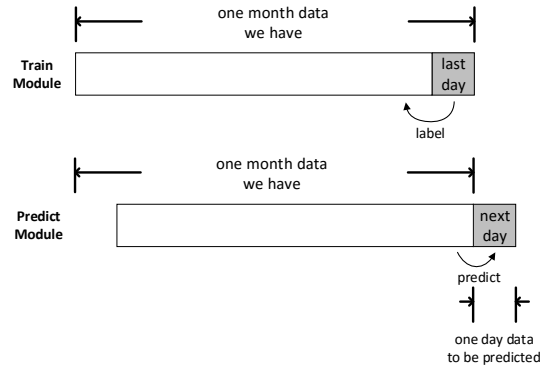


Fig. 2. Training Set and Test Set

- *Behaviors Statistics* stands for the count of a user's behaviors. The more this user browse, the higher possibility he will buy. There is an example in the left of Figure 3 to explain the definition. In this example, a user click for 3 times in the first day, 1 time in the second day and 2 times in the fourth day, so the count of his total behaviors in the last four day equals $3+1+0+2=6$.
- *Active Days* means the count of active days of a user. This feature could represent the positivity of a user directly. There is an example in the right of Figure 3 to explain the definition. In this example, a user login in the first day, the second day and the fourth day, so the count of his active days in the last four day equals $1+1+0+1=3$.

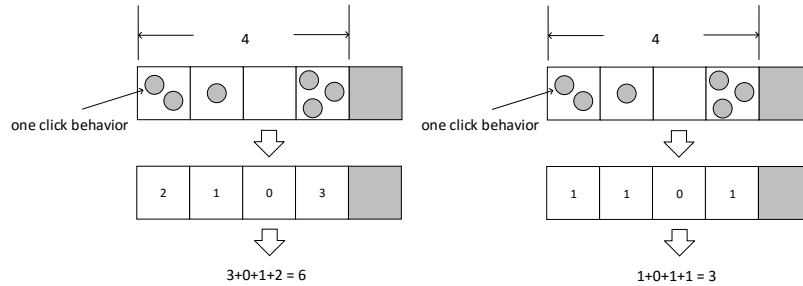


Fig. 3. An Example of Behaviors Statistics

Item Quality reflects the popularity of a item. Obviously, more popular items have bigger tendency to be bought.

- *Last Browsed Day* represents the last day a item is browsed.

- *Conversion Ratio* represents the ratio of purchase behaviors of a item in its total browsed behaviors.
- *Behaviors Statistics* stands for the count of a item’s browsed behaviors. The more this item is browsed, the higher possibility it will be bought.
- *Active Days* means the count of days a item is browsed. This feature could represent the popularity of a item.

Category Quality describes the popularity of a category. The definition of *Last Browsed Day*, *Conversion Ratio*, *Behaviors Statistics* and *Active Days* in it is similar with those in **Item Quality**.

User-Item Interaction describes the interaction between the user and item. it is a direct aspect to reflect the willing that the user want to buy the item.

- *Behaviors Statistics* represents the count of a user’s browsing behaviors on one item.
- *Active Days* means the count of days in which the user browse the item.

User-Category Interaction represents the interaction between the user and the category. It is similar with **User-Item Interaction**, and *behaviors Statistics* and *Active Days* will be generated in the same way.

4.3 Filtering Module

This purchase prediction task is based on a typical O2O business model, in which users pay online and consume offline. This means that users are not willing to buy items which is far away from them because they have to go there to consume. Based on the fact, we propose Filter Module to remove those pairs with too long distance. In detail, in a $(user, item)$ pair, if the distance between the location of the *item* and the *user* is bigger than L , any purchase behaviors will happen on this pair. We set $L = 100km$ from experience in this paper.

4.4 Reduced Data

Because the volume of our data set is too large, it will spend unacceptable time for training process in the machine learning approach. Hence, we use a reduced data set to solve this problem and keep prediction performance at the same time, which is showed in Figure 4.

Instead of using all the $(user, item)$ pairs happened in the one month, we use pairs show up in last n days to train and predict. $n = 1$ means that we use data in the last day while $n = 30$ means that we use all data over the whole one month.

5 Experiments

5.1 Data Description

The data contains two parts. The first part is the dataset D , the mobile behaviors data of users in the set of all items, with the following columns: *user_id*,

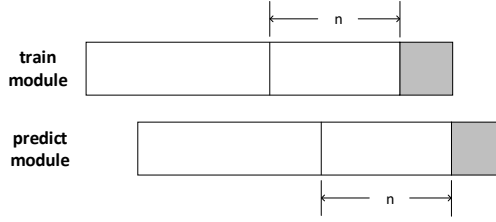


Fig. 4. Definition of the Reduced Data Set

item_id, *behaviors_type*, *user_geohash*, *item_category* and *time*. The second part is the dataset *P*, the subset of items data, with the following columns: *item_id*, *item_geohash* and *item_category*. The training data contains the mobile behaviors data of certain quantity of sampled users (*D*) from November 18, 2014 to December 18, 2014. The evaluation data is the purchase data of these same users of the items in *P* in December 19, 2014. Summary statistics of the data are listed in Table 1.

Table 1. The Statistics of The Data Set

#table <i>D</i>	#table <i>P</i>	#user set	#total item set	#sub item set	#category set
5,822,532,780	14,397,493	5,000,000	156,226,243	13,435,163	13,128

5.2 Two Rule-Based Baseline

CartRule is the first strategy of most participants, and we select it as our first baseline. In detail, *CartRule* thinks that if a user add a item into his cart and doesn't buy it in that day, it's likely that he will buy it in the next day. In addition, we propose *CartRuleTime* which adds time factor into consideration based on *CartRule*. *CartRuleTime* thinks that if a user add a item into his cart and doesn't buy it after *m* o'clock ($m \in \{0, 1, \dots, 23\}$) in that day, it's likely that he will buy it in the next day. When *m* is set to 15, the performance is the best according to our experiments.

5.3 Result

We set $n \in \{1, 2, 3, 4\}$ in the reduced data in this paper and apply three classifier: LR (Linear Regression), RF (Random Forest) and GBDT (Gradient Boosting Decision Tree). Table 2 shows the prediction performance of different approaches in this purchase prediction task. 'n1_LR' means that $n = 1$ and *classifier* = LR,

Table 2. Performance of Different approaches

Approach	F1(%)	Precision(%)	Recall(%)
CartRule	5.5480	4.1824	8.2377
CartRuleTime	5.8007	5.3018	6.4033
n1_LR	6.8985	7.8298	6.1652
n1_RF	8.0064	8.3249	7.7114
n1_GBDT	8.3841	8.2591	8.5129
n2_LR	6.3266	6.5820	6.0902
n2_RF	7.9830	8.5378	7.4959
n2_GBDT	8.4264	8.2036	8.6617
n3_LR	6.5542	7.4360	5.8593
n3_RF	7.9152	8.9481	7.0960
n3_GBDT	8.4719	8.6591	8.2926
n4_LR	6.5015	7.1628	5.9520
n4_RF	7.9152	8.8237	7.1764
n4_GBDT	8.4410	8.4336	8.4485
champion	8.78	8.76	-

‘n4.GBDT’ means that $n = 4$ and *classifier* = *GBDT*, others could be explained in the same way. *CartRuleTime* has a little improvement compared to *CartRule* because *CartRuleTime* takes the time factor into consideration. Those machine learning approaches we proposed have a much better performance than two rule-based methods, which could prove the effectiveness of our approaches to some extent. Compare the performance of different classifiers, we could see easily that *GBDT* is the best choice. With the increase of n , the F1-score changes littlely. This phenomenon proves that reduced data could accelerate the process of machine learning and keep the performance at the same time.

Table 3. Performance of Different Features

Approach	F1(%)	Precision(%)	Recall(%)
U+I+C	2.8068	7.3183	1.7364
UI+UC	7.4938	8.2366	6.8740
All	8.4719	8.6591	8.2926

To prove the effectiveness and robustness of feature families explored, we test a series of combination of feature families on *n3_GBDT*, which is the best result mentioned above. *U+I+C* means we use quality features only and ‘*UI+UC*’ means we use interaction features only. The performance of *U+I+C* is poorer than *UI+UC*, which explains the importance of interaction features. The performance of *All* is better than *UI+UC*, which reflects the supporting role of the quality features.

6 Conclusion

We present a machine learning approach to solve the purchase prediction task launched by the Alibaba Group. Five kinds of features are explored to describe the willingness of users' purchase desires on items. In particular, we take the time and spacial factor into consideration. Experimental results prove the effectiveness of our proposed approach.

Acknowledgement

This work was supported by National High Technology R&D Program of China (Grant No. 2015AA015403, 2014AA015102), Natural Science Foundation of China (Grant No. 61202233, 61272344, 61370055) and the joint project with IBM Research. Any correspondence please refer to Yansong Feng.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17(6), 734–749 (2005)
2. Ju, B., Ye, M., Qian, Y., Ni, R., Zhu, C.: Modeling behaviors of browsing and buying for alidata discovery using joint non-negative matrix factorization. In: *Computational Intelligence and Security (CIS)*, 2014 Tenth International Conference on. pp. 114–118. IEEE (2014)
3. Koren, Y., Bell, R., Volinsky, C., et al.: Matrix factorization techniques for recommender systems. *Computer* 42(8), 30–37 (2009)
4. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755), 788–791 (1999)
5. Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7(1), 76–80 (2003)
6. Lü, L., Medo, M., Yeung, C.H., Zhang, Y.C., Zhang, Z.K., Zhou, T.: Recommender systems. *Physics Reports* 519(1), 1–49 (2012)
7. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: *Proceedings of KDD cup and workshop*. vol. 2007, pp. 5–8 (2007)
8. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009, 4 (2009)
9. Wang, J., De Vries, A.P., Reinders, M.J.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 501–508. ACM (2006)