Ensemble of Diverse Models for Repeat Buyers Prediction

ZeYu Qiu, Jun Tian, Fan Tang
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
Beijing, China
qiuzeyu2013@ia.ac.cn, tianjun2013@ia.ac.cn, tangfan2013@ia.ac.cn

Abstract

This paper reports the approach of our team to predict repeat buyers after sales promotion. In order to reduce the promotion cost and enhance the return on investment, it is important to identify potential loyal customers from new buyers. To solve this problem, we first implement three existing methods, including Gradient Boosting Decision Trees (GBDT), Factorization Machine (FM) and Logistic Regression (LR), to mine various sides of information from data. Each of the individual models is optimized by extracting a set of appropriate features. Then we propose a rank-based ensemble method which greatly improves the result of our individual models. Our final solution obtains a score of 0.7094, giving us the fifth place in Repeat Buyers Prediction Competition of IJCAI.

1 Introduction

To attract more new buyers, merchants sometimes make big promotions on special dates. However, many shoppers just take it as a one-time deal, and these promotions may have little impact on sales. The Repeat Buyers Prediction Competition hosted by Alibaba Group aims to solve this problem. The task of competition is to predict whether a new buyer of a given merchant will purchase items from the same merchant again within 6 months after promotion in Double 11(Nov 11th).

The competition consists of two stages. In the first stage, the training set is consisted of 7030724 user action logs on 4995 merchants. The test set contains 7027944 user logs on the same merchants and shares the same format as the training set. In the second stage, the dataset maintains the same format as in stage 1 and the scale is about five times larger. In order to handle such big data, the organizers afford all participants a distributed development environment to run their algorithms. Each user log can be viewed as a vector (UserID, UserAge, UserGender, MerchantID, Label, ActiveLog). Moreover, the ActiveLog contains interactions between a user and a merchant range from May 11th to Nov 12th. Each interaction is described like (ItemID, BrandID, TimeStamp, ActionType). It is remarkable that the dataset contains not only label information about whether a user is a repeat buyer or not, but also many history action logs labeled with -1. The goal of the competition is to predict the probability that a new buyer of a merchant will buy again after promotion. The evaluation metric used in this competition is the Area under the Curve of receiver operating characteristic (AUC).

Clearly, the repeat buyer prediction problem could be viewed as a classification problem. Some unique features of this dataset make the problem a little different from a simple classification:

- The percentage of records which are labeled with 1 or 0 is 0.038%. Furthermore, the amount of negative records is about 15 times larger than the positive ones in training set. This huge sparsity is a big challenge to deal with.
- The training dataset and testing dataset are from the long-term user's behavior log in Tmall.com. The uncertainty and noise in the data set increase the difficulty of cleaning data and mining behavior patterns.

To solve these challenges, we extract many predictive features from both the labeled and unlabeled dataset. Based on these features, we implement FM, GBDT and LR in prediction task. In order to solve the unbalance problem of blending models, we develop an ensemble approach which utilizes the rank information to ensemble different models. By using this ensemble method, we obtained an AUC score of 0.70393 and reach the 4th place in stage 1. In stage 2, by leveraging the LR provided on the cloud platform together with features used in GBDT and FM we get a score of 0.7094, reaching the fifth place on the final leaderboard.

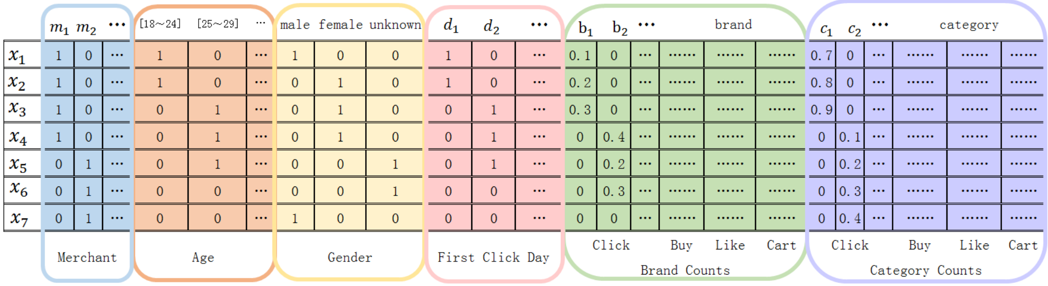
The rest of the paper is organized as follows. We introduce all the individual models with feature engineering and their per-formance in Section 2. In Section 3 we describe an approach to combine different models by rank information. Finally, we draw a conclusion in Section 4.

2 Individual Models

In this section, we introduce three models used in the competition respectively. The features and results of each model will also be described in detail.

2.1 Factorization Machines

Latent factor model and its variances are probably the most popular collaborative techniques that demonstrate significant performance in prediction problem[Koren, 2008],[Koren *et*



	m_1	m_2	⋯	[18~24]	[25~29]	⋯	male	female	unknown	d_1	d_2	⋯	b_1	b_2	⋯	brand			c_1	c_2	⋯	category		
x_1	1	0	⋯	1	0	⋯	1	0	0	1	0	⋯	0.1	0	⋯	⋯⋯	⋯⋯	⋯⋯	0.7	0	⋯	⋯⋯	⋯⋯	⋯⋯
x_2	1	0	⋯	1	0	⋯	0	1	0	1	0	⋯	0.2	0	⋯	⋯⋯	⋯⋯	⋯⋯	0.8	0	⋯	⋯⋯	⋯⋯	⋯⋯
x_3	1	0	⋯	0	1	⋯	0	1	0	0	1	⋯	0.3	0	⋯	⋯⋯	⋯⋯	⋯⋯	0.9	0	⋯	⋯⋯	⋯⋯	⋯⋯
x_4	1	0	⋯	0	1	⋯	0	1	0	0	1	⋯	0	0.4	⋯	⋯⋯	⋯⋯	⋯⋯	0	0.1	⋯	⋯⋯	⋯⋯	⋯⋯
x_5	0	1	⋯	0	1	⋯	0	0	1	0	1	⋯	0	0.2	⋯	⋯⋯	⋯⋯	⋯⋯	0	0.2	⋯	⋯⋯	⋯⋯	⋯⋯
x_6	0	1	⋯	0	0	⋯	0	0	1	0	0	⋯	0	0.3	⋯	⋯⋯	⋯⋯	⋯⋯	0	0.3	⋯	⋯⋯	⋯⋯	⋯⋯
x_7	0	1	⋯	0	0	⋯	1	0	0	0	0	⋯	0	0	⋯	⋯⋯	⋯⋯	⋯⋯	0	0.4	⋯	⋯⋯	⋯⋯	⋯⋯
													Click			Buy	Like	Cart	Click			Buy	Like	Cart
	Merchant			Age			Gender			First Click Day			Brand Counts						Category Counts					

Figure 1: An Overview of FM Feature

al., 2009],[Rendle, 2012b]. We will introduce how we deal with the repeat buyers prediction problem by using one of factor models.

Model Introduction

Factorization machines[Chen *et al.*, 2011] are general predictor working with any real valued feature vector. They can model all nested interactions between all features up to the specified order. It has been shown in [Rendle, 2010] that FM would mimic many of the most successful approaches for the task of collaborative filtering. Moreover, FM has already been proved effective in many competitions. So we design features and doing repeat buyer prediction by using FM.

A 2-way FM captures all single and pairwise interactions between variables. This makes FM doing well in prediction problems involving large categorical features, especially with sparse data. More importantly, the computation time of FM is linear time $\mathcal{O}(kn)$. This feature makes FM applicable to handle large scale data size problems. The second-order factorization machine is defined as:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \qquad (1)$$

Several learning algorithms have been proposed to solve for the parameters of FM including Stochastic Gradient Descent (SGD), Alternating Least Square (ALS) and Markov Chain Monte Carlo (MCMC). Among these algorithms, The MCMC method is the easiest to use because there is only one parameter of standard deviation to tune.

Feature Engineering

We simply consider the prediction problem as a binary classification problem. The dataset provides a rich of information about user's history logs, but most of them are unlabeled. To simplify the problem, only labeled user-merchant pairs are considered here. The reason why we can safely ignore unlabeled data is that labeled data is enough to describe a user's interest level towards a new merchant. All the features used in FM are listed as follows:

- **User Level:**
  - Age of the user,
  - Gender of the user
- **Merchant Level:**
  - Merchant ID.
- **Interaction Level:**
  - First visit day,
  - The brands/categories a user buys on a merchant,
  - The count of brands/categories that a user clicks/buys/likes/carts on a merchant.

All the id features above are considered as categorical variables and are encoded for FM with one-hot schemes. The count features are converted to log-scale and then be truncated. Different from the custom user-merchant matrix in recommendation systems, our features do not include the user id. The reason is that the users in training dataset have no intersection with users in testing dataset. The grouped features is shown in Figure 1.

Result

In this competition, we use the libFM[Rendle, 2012a] as our tool to make prediction. Our experiments show that the performance of MCMC method is much better than ALS or SGD method. By setting the initial deviation as 0.1 and the dimension as 1, 1, 4, we get a score of 0.682 on off-line training dataset. Then we train the model on the total training dataset in stage1 and reach a score of 0.686 on test dataset in stage1, which is a slight improvement.

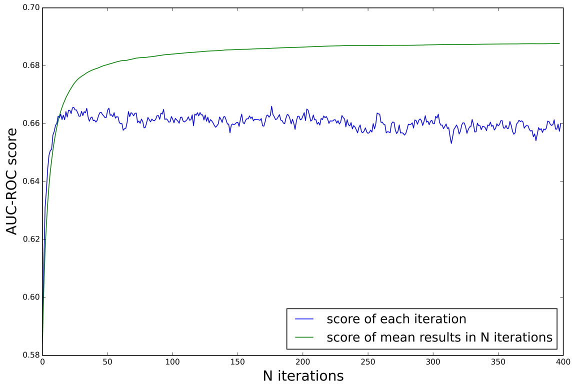


Figure 2: The Result of FM by Using MCMC

During our experiments, we also find that the number of iterations will influence the result. In MCMC method, the final prediction result is just the average results of all iterations. Figure 1 shows the result of each iteration and the average results. Considering that the parameters learned by libFM is not stable during the first 50 iterations. We decide to drop out the results of first 50 iterations. And this improvement promotes the score to 0.688.Due to the platform limitation in stage 2, we can't apply the FM for prediction. The training and testing are separated on the distribute computation platform, making it really diffcult to exchange hundreds of thousands of parameters.

2.2 Gradient Boost Decision Tree

Boosting is a general method for improving the accuracy of any given learning algorithm[Jahrer *et al.*, 2012]. We will describe the details of how we use boosting method to predict repeat buyers.

Model Introduction

The Gradient Boosting Decision Trees (GBDT) [Friedman, 2001] is one of the most effective machine learning models for prediction tasks. In recent years, we have witnessed a good number of winning solutions that contain GBDT as a crucial component. GBDT is typically trained with decision trees of a fixed size as base learners. As an ensemble method, GBDT is very robust to outliers and can handle missing fields. Also, it is good at discover the potential interactions between different features automatically by selecting partial features.

Feature Engineering

Extracting effective features is one of the most important things in using GBDT[Wu and Ferng, 2012][Wang and Chen, 2011]. By selecting only part of features to build up the base decision trees, GBDT can highly improve the model's robustness. However, unlike polynomial regression, GBDT is weak in mining the interaction information between features. Therefore, how to design predictive features is a major task in the competition. Based on features' characteristics, we group them into the two parts: original features and synthetic features. The considerations when designing features are described below.

Different types of users have different kinds of behaviors. For example, females like to spend more time before and after they buy something than males. And males are more purposeful when they visit some merchants. According to our statistics and analysis, we put forward a set of continuous features to describe user behaviors and merchant attributes. These features are called original features, such as the number of sellers each user visits, the number of sellers each user visits, total counts of a merchant and so on.

Except for original features, we also have extracted some synthetic features. Considering those people who like to visit merchants on-line frequently but rarely make deals on-line, the original features cannot catch this kind of information. So we extend our features based on original features. The main part of original features and synthetic features are listed as follows

- **Original Features:**
  - *nClickSellers*: the number of sellers each user visits.
  - *nRepeatSellers*: the number of merchants each user repeat buys.
  - *nClicks*: total click number of a user.
  - *avgClicks*: the average number of clicks on a merchant per consumer.
  - *avgBuys*: the average number of purchases on a merchant per consumer.
  - *nRepeatBuyUsers*: the number of repeat buyers of a merchant.
  - *nClickUsers*: the number of users who visit a merchant.
  - *nClicks*: the total clicks of a merchant.
  - *avgClicks*: the average clicks per consumer of a merchant.
  - *avgCounts*: the average buy/click/cart counts per item/brand/category.
  - *nCclicks*: the number of brands/items/categories that a user clicks on promotion day.
  - *mBuys*: the number of brands/items/categories that a user buys on promotion day.
- **Synthetic Features:**
  - *repeatBuySellersRatio*: the ratio between nRepeatBuySellers and nClickSellers.
  - *repeatBuyUsersRatio*: the ratio between nRepeatBuyUsers and nClickUsers.
  - *nClickBuyDiff*: the number of clicks minus the number of buys.
  - *buyDays*: the total days that a consumer spends on a merchant and finally buys something.
  - *buyClickRatio*: the count of buyers divided by the count of click users.
  - *firstActDay*: the first click day before promotion day.

Result

We use GBDT model afforded by the scikit-learn package. The optimization goal of GBDT in this competition is "exponential" which recovers the Adaboost algorithm. Since a GBDT model is built sequentially by using weak decision tree on sampled data, other parameters are heavily related to decision tree. To avoid over-fitting, we set the max depth of each tree as five and set the minimum number of samples required at leaf as 200. During training, every tree only randomly selects about 120 features to make decisions. Because the GBDT cannot be trained in parallel, we use only 200 trees and reach a score of 0.693 in stage1.

In stage 2, we used the GBDT provided by organizers. The score of AUC is 0.646, which is much lower than the result we get on stage 1. The most likely reason is that the GBDT on the cloud platform has a different implementation with scikit-learn package. The different optimization goal and unsuitable features lead to the difference between stage 1 and stage 2.

We have tried to implement a different GBDT model on the cloud platform through different methods. One of the best is to save the GBDT models trained in stage 1 into a text file and then upload the file to the cloud platform for online prediction. And this method gets a score of 0.702 on stage 2. It shows that the GBDT model trained in small data performs at least the same as ones trained in big data.

2.3 Logistic regression

Logistic regression is widely used for binary classification in industry. In this competition, we use LR to model the relationships between different features and predict the probability that a buyer will become a repeat buyer. As mentioned before, the features in FM are all category features and in GBDT are all continuous features. Thinking of that LR can easily handle both continuous and category features, we decide to use LR as a feature level blending.

In stage 1, we only get a score of 0.684 using LR which is slightly lower than FM and GBDT. In stage 2, we use the LR provided by the cloud platform. After combining features of FM and GBDT, the result is much higher than stage 1, which reaches 0.708 on the leaderboard. Then by using up-sampling of positive data, we extend the positive data to three times larger. Finally we get a score of 0.7094 on the leaderboard.

3 Ensemble

To generate a better performance, we choose to blend the results of all the individual models discussed in Section 2. Apparently, it is easy to combine the results by linearly merging all the results. But the probabilities generated by base models have a wide range. The maximum probability output by GBDT is almost 1.0 and that of FM is less than 0.5. Thinking of that the AUC is decided by the relative rank of each user-merchant pair, we propose a rank based blending method to fix the imbalance between different individual models. Assuming that $rank_n(i)$ denotes the rank of an instance i sorted by the probability generated by model n, the final result could be calculated as:

$$prob_i = \frac{1}{n} \sum_{k=1}^{n} \frac{w_k}{rank_k(i)} \tag{2}$$

where w denote the result of the i instance in one model.

Table 1: My caption

Stage	Model	Score
Stage 1	LR	0.6826
	GBDT	0.7013
	FM	0.7003
	Linear Ensemble	0.7036
	Ensemble	**0.7028**
Stage 2	LR	**0.7094**
	GBDT(offline)	0.7021
	GBDT(online)	0.6432

In stage 2, due to the limitation of the cloud platform, we can't merge the results from LR and GBDT provided by organizer. So we only test our method in stage 1. Table 1 shows all of results we get. And it could prove the effectiveness of our rank-based blending approach.

4 Conclusion

In this paper, we have introduced our methods dealing with the Repeat Buyer Prediction competition in ICJAI. Our basic intuition is to try different models to mine varying aspects of data. Then we propose three individual models including FM, GBDT and LR to predict the probability that a user will become a repeat buyer. Our experiments show that they all have a good performance. We further propose a rank-based ensemble method to blend the results of individual models.

5 Acknowledgments

We would like to acknowledge the organizers for holding this interesting competition. We also thank Alibaba Group for providing cloud platform and dataset. Finally, we want to thank the help afforded by seniors.

6 References

References

[Chen *et al.*, 2011] Tianqi Chen, Zhao Zheng, Qiuxia Lu, Weinan Zhang, and Yong Yu. Feature-based matrix factorization. *arXiv preprint arXiv:1109.2271*, 2011.

[Friedman, 2001] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[Jahrer *et al.*, 2012] Michael Jahrer, A Toscher, JY Lee, J Deng, H Zhang, and J Spoelstra. Ensemble of collaborative filtering and feature engineered models for click through rate prediction. In *KDDCup Workshop*, 2012.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[Rendle, 2010] Steffen Rendle. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 995–1000. IEEE, 2010.

[Rendle, 2012a] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[Rendle, 2012b] Steffen Rendle. Social network and click-through prediction with factorization machines. 2012.

[Wang and Chen, 2011] Chieh-Jen Wang and Hsin-Hsi Chen. Learning user behaviors for advertisements click prediction. In *Proceedings of the 34rd international ACM SIGIR conference on research and development in information retrieval Workshop on Internet Advertising*, pages 1–6, 2011.

[Wu and Ferng, 2012] Kuan-Wei Wu and Chun-Sung etc. Ferng. A two-stage ensemble of diverse models for advertisement ranking in kdd cup 2012. *KDDCup*, 2012.