The Hierarchical Model to Ali Mobile Recommendation Competition

Suchi Qian, Furong Peng, Xiang Li, Jianfeng Lu
School of Computer Science and Engineering, Nanjing University of Science and Technology
Nanjing, China
E-mail: qiansuchi2012@gmail.com, pengfr@njust.edu.cn, implusdream@gmail.com, lujf@njust.edu.cn

***Abstract*—Recommendation Engines have gained the most attention in the Big Data world. In order to promote the application of big data, Alibaba Group organized the big data recommendation competition, which provides the big data processing platform and one billion behavior records to participants. The competition requires the participants to learn the model from the user's behaviors within one month and then predict the purchase behavior in the following day. There are four kinds of different behaviors included: browse, add-to-cart, collection and purchase. The F1-score is as the metric to evaluate the performance.**

Finally，our team achieves the top score of 8.78%, and our success can be owed to the following aspects: First, we model the recommendation problem as the binary classification problem and design the hierarchical model; Second, in order to improve performance of single classifier, we adopt the sample filtering strategy to select valuable samples for training, which not only boosts the performance but also speeds up the training; Third, the classifier fusion strategy is used to improve the final performance.

This paper details our hierarchical model and some relevant key technologies adopted for this competition. This hierarchical model is also the framework of data processing, which is composed of four layers: 1) Sample filtering layer, which removes a large number of invaluable samples and reduces the computing complexity; 2) Feature extraction layer, which extracts extensive features so as to characterize the samples from all possible views; 3) Classifying layer, which trains several classifiers by different sampling strategy and feature groups; 4) Fusion layers, which fuses the results of different classifiers to obtain the better one. Our score in competition manifests the reasonableness and feasibility of our model.

Keywords—big data; recommendation;competition;hierarchical model; fusion

I. INTRODUCTION

The era of big data has come and will profoundly change our society. Due to its huge commercial value, it not only attracts attention from academic community, but also from industrial community. Big data means massive data sets containing large, various and complex structure, which is difficult to store, analyze and visualize. With the increasing of new technology, products and application, the amount of data appears rapid growth. Researches show that data are produced double by human every two years. However, although data will increase 50 times within the next decade, the number of specialists who can handle those data only increase by 1.5 times[1,6,7,8,9]. Due to its various properties like volume, velocity and variety, variability, value and complexity, Big data causes many challenges. Compared with traditional data analysis, it requires new technologies and architectures to capture and analyze its value.

When facing too much data, people find that it is difficult to obtain useful information sometimes. To deal with the information overload problem, recommender system is proposed which provides user's interesting information such as movies, books, music, bookmarks, CDs, news, images and TV programs[1,2,3]. For e-business system, when customers browse products online, many famous companies provide recommending service so as to save customers' time, such as Amazon and Taobao online store. To promote the development of recommender system, many competitions have been organized. From 2006 to 2009, Netflix, Inc. organized the competition of recommending movies with the prize of 1 million dollars. The Hetrec2011 competition was held in conjunction with the 5th RecSys conference. Alibaba group also organized the competition of Tmall Recommendation Algorithm in 2014. With the increasing requirement for recommendation technology, a lot of scientists and engineers pay more attentions to the recommendation system and great progress has been made. In the past decades, many recommendation algorithms are proposed [1,3,4,5], and they are generally divided into three categories: content-based, collaborative filtering (CF) and hybrid methods.

In order to promote the development of big data processing in industry, Alibaba Group organized the big data recommendation competition in 2015 also named Ali Mobile Recommendation Algorithm (AMRA). Final competition is hold on Yunshanfang platform, which is one of cloud platform for the big data processing and operated by Alibaba group. Due to its challenge and high prize (RMB 300K), more than 7000 teams have attended this competition.

After analyzing the raw data and reviewing related work, we adopt the following strategy: First, select some existing algorithms by comparison; Then, boost the performance of single algorithm as possible by training; Finally, fuse the result from each algorithm so as to achieve the better performance. Based on this general strategy, we design the following technical schema: First, recommendation task can

1.Yushanfang: http://yushanfang.com
2.Aliyun: http://www.aliyun.com

978-1-4673-8493-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDMW.2015.75

be modeled as the binary classification problem; Then, we design the hierarchical model with four layers; Third, after analyzing the sample set, sample filtering strategy is adopted so as to remove a large number of invaluable samples and just remain the valuable samples for training, which can improve the performance of classifier and speed up the training; Finally, some different classifiers and fusing strategy are tried, and ensemble averaging based on the stacking GBDT[20] is as the final one. The final score indicates that our strategy is reasonable and practical.

The rest of this paper is organized as follows: Section II will briefly introduce the competition from multi-aspect; Section III is the overview of the proposed model; Section IV will describe our sample filtering strategy; Section V will detail the extracted features; Design of classifier will be presented in Section VI; Fusing strategy of classifier is given in Section VII; Section VIII will conclude the paper.

II. Brief Introduction of AMRA

AMRA is the competition based on the real user-commodity behavior data on Alibaba's mobile-commerce platforms, and the user's historical behaviors are provided to participants. A transaction log records the associated commodity, user's location, time, and behavior type. The participants need to predict the actions happened in the following days. Specifically, given the behavior records from Nov. 18 to Dec. 18, the participants are required to predict the behaviors happened in Nov. 19. The prize for Champion, First runner-up and Second runner-up is RMB300k, 50k, 20k respectively.

Due to its great challenge and grand prize, AMAR has attracted more than 7000 teams to participate. Next, we'll introduce the rules, data and evaluation metric.

A. Brief Introduction of Rules

The competition consists of two seasons, season 1 & 2. The period for season 1 is March 20-April 25, and top 500 teams can enter Season 2. Season 2 is from April 30 to July 1. In season 2, the data are only available on the Yushanfang platform, and all the teams must submit their results on that platform.

Season 2 are divided into two parts:

1) Part 1, April 30 - June 23. By the end of Part 1, teams among the top 200 will enter Part 2.

2) Part 2, June 24 - July 1. By the end of Part 2, teams among the top 5 will be invited to participate in the final presentation.

Part 1 answers shall include the purchase data of 50% of all users who have purchase behavior on the very day of observation.

Part 2 answers shall include the purchase data of 100% of all users who have purchase behavior on the very day of observation, namely the users in the stage of Part 2 is twice that of Part 1.

Participants are required to log in the Tianchi Platform to access and use the massive Taobao data, debug their models with Map&Reduce, and submit the results, where SQL and machine learning algorithm packages has been integrated into the platform,.

In season 2, system will conduct the evaluation once per day,and the ranking list will be updated every day according to F1 scores.

B. Brief Introduction of Data

During season 1, Alibaba Group released 10,000 users' behavior data and a million product information at the preliminary. The goal of the competition is to use a month's training data (from Nov. 18 to Dec.18) to predict the following day's (Dec. 19) purchase results . The dataset is a small subset, which can be downloaded by the participant's computer. The participants design the algorithm in their own computer and submit the result to the platform for evaluation. In this paper, we mainly focus on the algorithms on season 2 which is the real big data challenge.

The datasets provided by the organizer are described as follows:

U: the whole user set (ID)

I: the whole item set (ID)

P: item subset (ID) $P \subseteq I$

D: the whole user behavior dataset

Alibaba Group provides the training sets as described in following two tables.

TABLE I. THE TABLE OF TIANCHI_MOBILE_RECOMMEND_TRAIN_USER(5.8 BILLION ROWS)

Column	Description
User_id	User identity
Item_id	Item identity
Behavior_type	1:Browse 2: Collect 3: Add to cart 4: Purchase
User_geohash	User location (can be null)
Item_category	Commodity's category
Time	Action time

TABLE II. THE TABLE OF TIANCHI_MOBILE_RECOMMEND_TRAIN_ITEMN(14 MILLION ROWS)

Column	Description
Item_id	Item identity
Item_geohash	Item location (can be null)
Item_category	Commodity's category

C. Brief Introduction of Platform

During season 1, the participants can design the algorithm on their own computer and submit the result to the platform for evaluation.

During season 2, Yushanfang, the big data platform, can be accessed for all participants. The participants build the model online and output the prediction based on above two tables.

D. Brief Introduction of Evaluation Metric

The evaluation metric for this competition is the classic precision, recall and F1-measure, which is defined as follows.

$$precision = \frac{|\,prediction\ \ set \cap reference\ \ set\,|}{|\,prediction\ \ set\,|} \tag{1}$$

$$recall = \frac{|\,prediction\ \ set \cap reference\ \ set\,|}{|\,reference\ \ set\,|} \tag{2}$$

$$f1 - score = \frac{2 \times precision \times recall}{precision + recall} \tag{3}$$

Where, prediction set is the submitted purchase data and reference set is the real purchase data. F1-score is as the only standard of the final evaluation.

III. Overview of our Hierarchical Model

Through data analysis, we find that the behavior of user's purchase can be classified into two categories: the non-interactive pair that does not have any interaction between user and item before the user action date, and the interactive pair that has some kinds of interaction before the date. The rate of interactive pairs and non-interactive pairs accounts for 30-40% and 60-70% respectively in the reference set. Although the non-interactive pairs dominate the purchase actions, the number of interactive pairs is still too large (7×10^{16}) to be processed. The number of purchase actions in test day is only 160k. Therefore, what we should do is to predict the purchase actions from the interactive pairs, namely, find 1.6×10^{5} positive samples from 7×10^{16} samples, which is a challenging problem Next, problem definition and our model will be introduced first.

A. Problem Definition

The recommendation task can be modeled as the binary classification problem in our solution. In the dataset, the sample set is denoted as $\mathcal{U}$,and each sample can be represented as a triple (U, I, T), which is composed of user id U, item id I and the action date of user behavior T, where T is called **sample-date**. As the non-interactive samples are very difficult to predict and process, we only consider to predict from the interactive samples. The date when user interacted with item before sample-date is called **interaction-date**. The sample whose user and item have interaction (e.g. browse, collection, add-to-cart and purchase) before the associated date is called interactive sample. The interactive sample set is denoted as $\mathcal{S}, \mathcal{S} \subseteq \mathcal{U}$. A sample s_i is labeled as positive ($y_i = 1$) if the user buys the item in the sample-date, and negative ($y_i = 0$) otherwise. The label set is defined as $\mathcal{Y}$.

The aim of the competition is to predict the purchase pairs(U, I) happened in Dec. 19. For convenience, we map the Nov. 18 to Dec. 19 into 0-31. As only the interactive samples are considered, we just predict the label of interactive samples whose date is 31th. Therefore, all the samples whose date is 31th are in our testing set $\mathcal{S}_{test}$ (T=31). The training set $\mathcal{S}_{train}$ includes those samples whose date is before 31th(0<=T<=30).

In the training set, all the labels of training samples are available. However, the labels of testing samples are always unknown, which are used to evaluate the participant's result by organizer. We train the model $f : \mathcal{S}_{train} \rightarrow \mathcal{Y}_{train}$ from the training set, and then use f to predict the labels of $\mathcal{S}_{test}$. The more correctly the model predicts, the higher F1 score will be.

B. Hierarchical Model

Based on the problem defined in the previous sub-section, we generally introduce the hierarchical model in this section. Each layer of the hierarchical model will be discussed in the following sections. The overview of the designed model is described in figure 1.

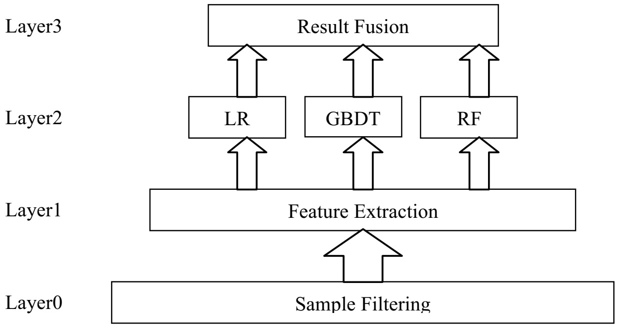

Fig. 1.   Overview of our hierarchical model.

The function of each layer is described as follows.

- Layer 0: The main function is to remove most invaluable samples (including training and testing). Through data analysis, we find that if time interval between interaction-date and sample-date is too long, the sample will be with very low positive probability. If these samples are removed, the performance can be improved significantly.
- Layer 1: This level should extract the useful features. The historical information of user-item pair is not well organized for classifiers, therefore, we reorganize these data and extract all possible features from different views. The extensive features can characterize the samples and will be provided to classifiers.
- Layer 2: Classification with different classifiers is carried out in this layer. Classifiers such as random forest (RF)[18], gradient boosting decision tree (GBDT) [10] and logistic regression (LR)[17] are trained in this layer, and all features are used to train, but, the simple combination of the training models usually cannot achieve high performance. However, if

we choose different subset of samples and different subset of features to train diverse classifiers, then fuse those models, which may improve the performance greatly.

- Layer 3: The main purpose is to fuse the results by the diverse classifiers to obtain better performance. Several strategies are tested. Experiment shows the hierarchical model can achieve the better performance by averaging the probability from all classifiers, including stacking GBDT mentioned in section VII.

There is no feedback loop in our hierarchical model, it's forward model and each layer plays the important role. In the following sections, we'll detail the function of each layer.

IV. Sample Filtering

Before extracting the features, we'll overview our task first. It's found that the number of negative samples is dominant compared with that of the positive samples. We need to design a filtering strategy to remove most of invaluable samples. For the convenience of description, the interactive samples are classified into three different categories: browse-sample, cart-sample and collection-sample. The samples, whose user has browsed the associated item, are called **browse-samples**. If the user has added the item to cart or collected the item, then the associated sample is called **cart-sample** or **collection-sample**.

Based on the data analysis, it's found that the user who has not interacted with an item for a long time, she/he will buy the item with very low probability. Based on the time span between sample-date and interaction-date, we can design an efficient filter. To exam the impact of date interval from interaction-date to the sample-date, we plot the recall of three different types of interactive samples in fig.2, where the date interval ranges from 1 day to 10 days. The samples, whose sample-date is 30, are used as the ground truth to compute the recall. From fig. 2, it can be seen that the recall decreases very quickly with the increasing of date interval. When date interval becomes larger than 7, the recall is smaller than 1%. We also plot the number of the interactive samples over the date interval in fig.3. and fig.4.

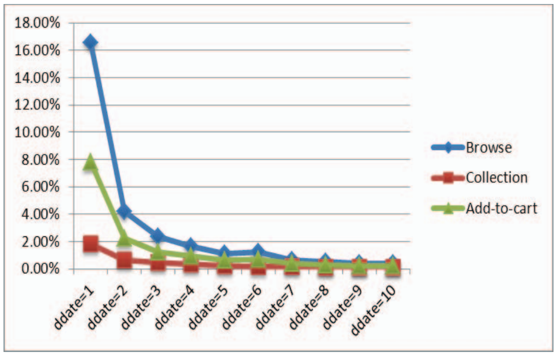

Fig. 2. The recall of different behaivors over date interval.

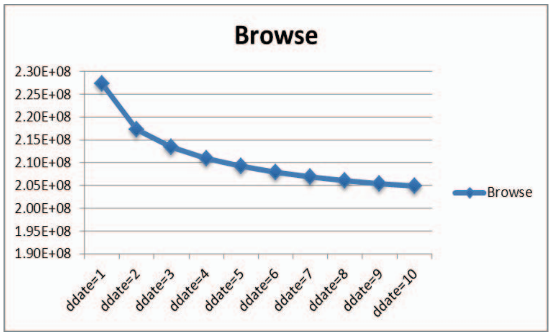

Fig. 3. The number of browse samples over date interval.

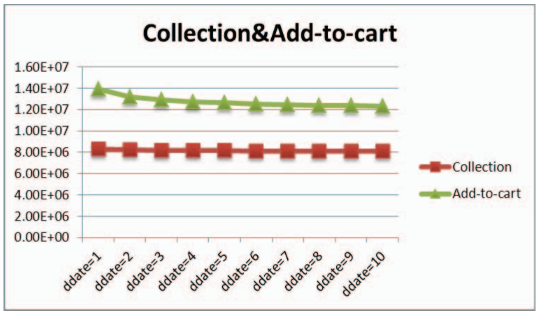

Fig. 4. The number of collection and add-to-cart samples over date interval.

The number of browse-samples is very large (0.2 billion) even if the date interval is 1 day. The recall of browse-samples decreases dramatically when the date interval increased from 1 day to 2 days. However, the number of browse-sample is still very large. Large number of samples will make our training and testing very slow. For this reason, we only consider the browse-samples whose date interval equals to 1 day in the training samples and testing samples.

Compared with the number of browse-samples, the number of cart-samples and collection-samples is small. We only adopt the cart-samples and collection-samples whose date interval is less than 7 days in the training samples and testing samples. There are two reasons for this strategy: 1) seven-day is a nature cycle for people's work and rest, hence, the behaviors may be repeated periodically; 2) the recall of samples, whose date interval is more than 7 days, is very small. The experimental results have also shown that the strategy of our sample-filtering is better.

V. Feature Extraction

After sample filtering, the real training set is determined. By analyzing the business, we discover ten categories of features that are helpful to predict the behavior of purchase.

TABLE III. Ten kinds of features

Features	Description
UI	User-Item features
U	User features

I	Item features
C	Category (Items) features
UC	User-Category features
GEO	Item Geography features
IC	Item-Category Ranking features
UIU	User-Item-User Ranking features
UCU	User-Category-User Ranking features
UIUC	User-Item-User-Category Ranking features

A. UI Features

As shown in the above table, User-Item features (UI) mean the relationship between user and item. If a user wants to buy something, he will first view the attribute and the price of a commodity and then add it to collection or cart so as to make comparison, which means that the user may buy this product in the future. In general, the more interaction between the user and the item, the more likely the user buys it.

For example, we count the total number of times that the user browses the item and the number of add-to-cart, collection and purchase respectively. Then, we introduce the weight features. According to the proportion of browse, collection, add-to-cart and purchase, we estimate the weight of four behaviors in the interaction. After that, we put the weighted sum of four kinds of behavior as our UI weight features. In short, UI features reflect the degree of user's preferences for an item.

UI features are the basis of all and its effect seems best in our experiments, after all, the essence of this problem is to predict the UI pairs .

B. U Features

User features (U) reflect the user's habits and we discover that users can be divided into several types based on different habits. For instance, some users buy the item immediately after browsing; some users like adding items to collection or cart before they make payments. What's more, some users buy a lot because they like shopping online very much. In a word, it can be estimated from the big data that users are active or not. In addition, there are some users who prefer to shop on weekend(Friday evening, Saturday, Sunday).

User features can distinguish whether a user is active or not by counting the frequency of the user's purchase quantity or purchase conversion rate. The number of items bought by a user is mainly used to determine whether the user is keen on shopping online. Meanwhile, the user conversion rate is mainly used to judge how many times they have interaction (including browsing, collection and add-to-cart) with items before they buy them. Besides, the purchase quantity per day is used to calculate week distribution of user purchase.

User features are effective information for our algorithms.

C. I Features

Item features (I) refer to the popularity of items. It is said that sales volume of items determines the selection of the user. If one user wants to shop online, he/she may do some searching work. In Taobao.com, there are many indexes for sorting the search results, which are usually related to the historical information of items, e.g., click rate, collection, comprehensive sales, etc. As a result, items with high historical purchase record will be sold better.

We calculate the frequency of the item purchase, the conversion rate of items, the popularity of items and the amount of items purchase per user. We use the weighted sum of browsing, collection, adding-to-cart and purchase to calculate the popularity of items like UI features. We estimate the search logic of Taobao.com by this way.

The final results of the experiment show that item features improve our score and best-selling items.

D. C Features

The data provided by Alibaba Group contains an "item category" column, each of which has an only category identity. C features refer to the popularity of item categories. For example, FMCG (Fast Moving Consumer Goods) will soon be run out of and twice purchase appears. However, the sales volume of luxuries will be small as few could afford them.

The methods of counting the conversion rate and weighted sum are also be adopted to judge the popularity of category.

However it surprises us that the item category features do not bring much improvement. The official stated that the items in the data are purchased online but used offline, so that the item categories are not quite different and the effect of category features is not obvious.

E. UC Features

According to prior knowledge, users tend to choose one of the item categories. User-Category features (UC) mean the relationships between user and item category. Users may have rigid requirements on some categories of items. It is assumed that a user has a car, and then he must buy car wash tickets. Based on it, the main purpose of this user is to buy necessities for his car. Item category of user's demand also follows the distribution of the day of a week. For example, if users like to watch movies, maybe he will buy movie tickets online every Friday.

Because of the user's rigid demand of one specific item category, we count the quantity of the user's demand and it will be reflected in the UC features. At the same time, we calculate the user's history demand for each category of items.

UC features conform to user's logic of shopping online and the effect is remarkable.

F. GEO Features

Item Geography features (GEO) are easy to understand. The data provided by Alibaba Group contains "Item_geohash" column and "User_geohash" column that can provide the location of items and users. We calculate the true distance by one function given by the official and find mobile users prefer to buy closer items. If online purchase and offline consumption items are the same except for the geographical position, users always buy items close to them. Large stores usually put their geographic location online. If a store doesn't mark geographic information, it is usually a small store or it does not pay much attention to online business. Users tend to choose large stores to shop.

Unfortunately, in the data provided by Alibaba Group, most users and items don't have geographic information because the mobile phone's positioning function is closed, so the effect of GEO features is not obvious.

G. IC Features

Item category features (IC) are rankings of the items in the same category. Through the rules of Taobao.com and user's habit, the user decides to pay for the items ranked front. In general, users prefer to buy items with higher popularity rather than others.

There will be a ranking of items in Taobao.com. We use the history data to estimate the rank of items in the same category. We consider not only item collection, add-to-cart and purchase, but also item popularity to estimate accurate ranking list in Taobao.com.

Since we use the training data to estimate the search system of Taobao.com, IC features improve our score.

H. UIU Features

User-Item-User Ranking features (UIU) are also kinds of ranking. In a period of time, UIU features reflect the importance of the item for the user. If users want to shop, users may make a list of items in the order of importance. According to the popularity of item, users will make a "list" in the mind to decide which one deserves to be bought. In other words, users rank all the candidate items and finally decide to buy one or several.

Also, through counting frequency, we can roughly estimate the "list" of the user by the history data of browse, collection, add-to-cart and purchase.

I. UCU Features

Broadly speaking, the ranking "list" of the item category for the user exists and we call it as User-Category-User Ranking features (UCU). This is a category ranking number in the user's "list" and also embodies the rigid demand of users.

In the same way, we calculate the frequency ratio to represent UCU features.

J. UIUC Features

User-Item-User-Category Ranking feature (UIUC) is a list of ranking. Compared with IC features, UIUC features pay more attention to user's real experience. For a real user, there is a list in user's mind that indicates the ranking of items in the same category that will play a decisive role in the purchase. We make full use of the factor that each user's preference of the item in the category to recommend personalized item for them.

The list is estimated by comparing the frequency ratio of the item that users interact with in the same category with the other item's frequency ratio.

K. Time Window

The influence of features will decay as time goes on. So we choose the time point. We just calculate features of d days before the test date. We set d as 1, 2, 3, 4, 5, 6, 7, 10, 15, 21, 30 in order to get more historical information.

By combining these ten kinds of features, we build L1 layer. The L1 layer basically consists of our prior knowledge, which reflects our understanding of business rules. Each sample is composed by user id, item id and action date of action behavior, UI features, U features, and other kinds of features. The set is defined as X . In the same way, we define two sets of X_{train} , X_{test} .

VI. Design of Single Classifier

The performance of single classifier is the foundation for the final fusion, which should be high as possible, so, different classifiers are tried. As the training data for predicting is unbalanced, the sampling strategy is used. Different sampling rate will generate different results and training classifier on different features will also generate different result. Before introducing the training strategy, we'll first introduce following 3 classifier: logistic regression (LR), random forest (RF), and GBDT.

A. Logistic Regression

Logistic Regression (LR) is a kind of simple and classic method for statistical learning and machine learning. We set Y as a random variable of predict results, $X = (x_1, x_2, ..., x_n)$ as an input of logistic regression. If the users has behavior of purchase, Y = 1, otherwise Y=0. The conditional probability $P(Y = 1 | X)$ means the probability of events occurrence with independent variables X. 1-P means the probability of events not happened. The logistic regression model is the conditional probability distribution as below:

$$P(Y = 1 | X) = \exp(w \cdot x + b) / (1 + \exp(w \cdot x + b)) \quad (4)$$

$$P(Y = 0 | X) = 1 / (1 + \exp(w \cdot x + b)) \quad (5)$$

Where X is the input, Y is the output, $w = (w_1, w_2, ..., w_n)$ is called the weight vector, b is called the offset vector, and $w \cdot x$ is the inner product of w and x .

For a given input instance X , we can calculate $P(Y = 1 | X)$ and $P(Y = 0 | X)$ through above formula, then, classify X into the class with high probability. We usually use gradient descent method or the quasi-newton method to estimate the logistic regression parameters. Following parameters in LR model should be considered: regularization type, maximum number of iterations and the iteration terminating condition.

We use the LR model (parameters: regularization type=2; maximum number of iterations=500; iteration terminating condition= 1.0e-06) and our features from Layer 1 to predict the results. According to our offline test, the F1-score of LR model is 7.00%.

B. Random Forest

Random Forest (RF) is an ensemble learning method for classification, regression and other tasks. Decision tree is a popular method of machine-learning. In particular, trees that grow very deep tend to learn highly irregular patterns, but they'll overfit training sets, because they have low bias and very high variance. Random forest is a way of averaging multiple deep decision trees, trained on different parts of the same training set with the goal of reducing the variance. And Random forest can greatly boost the performance. There are three main parameters in RF model: tree numbers, depth of tree and max record size.

We use the RF model (Parameter: tree numbers=800; depth of tree=15; Max record size=100000) and features from Layer 1 to predict the results. According to our offline test, the F1-score of RF model is 7.30%.

C. Gradient Boosting Decision Tree

Here, we'll introduce the model with highest score among all of our single models. Gradient boosting decision tree (GBDT) is a machine learning technique for regression and classification problem. GBDT produces a prediction model that is ensemble by weak prediction models called decision trees.

It's necessary to understand the concept of "Gradient Boosting". At each stage $1 \le m \le M$, it may be assumed that there is some imperfect model F_m . The performance of gradient boosting algorithm is improved by building a new objective function instead of changing the F_m . The new function is $F_{m+1}(x) = F_m(x) + h(x)$. The new estimator $h(x)$ can make model stronger and stronger. But how to get $h(x)$? $h(x)$ can be expressed as $h(x) = F_{m+1}(x) - F_m(x)$. Here, $h(x)$ is used to fit the residual $F_{m+1}(x) - F_m(x)$, each $F_{m+1}(x)$ will learn to correct its prediction F_m. Generally, loss function $\frac{1}{2}(F_{m+1}(x) - F_m(x))^2$ will be used. So, "Gradient Boosting" is a typical gradient descent algorithm. There are five main parameters for GBDT: tree number, depth of tree, shrinkage (learning rate), sampling ratio and feature ratio.

The main function of above parameters are as follows: in general, the more tree number means the better results; each single classifier with deeper depth (one decision tree) will own better performance; shrinkage, sampling ratio, feature ratio are used to resist over-fitting.

D. Training of models

Limited by the times of online submission, we use the samples whose sample-date is 30th as the ground truth to conduct the offline test. The ratio of positive samples and negative samples is about 1:100. This is typical unbalance classification problem, so, we down sampled the negative samples. After that, the ratio becomes 1:10, therefore, both the performance and processing speed is improved.

We trained LR, RF and GBDT classifiers. The best result of each model is listed in Table IV. GBDT outperforms the other two classifiers on the offline test. The result of GBDT is submitted for online evaluation and its online result is also shown in Table IV.

We also find the results of GBDT model with different parameters have some difference. Then we use different sampling ratios (1:10, 1:12 and 1:14) and parameters to get the diverse results of GBDT model, that is to say, we have several results of GBDT model with different setting so as to provide to the next step of model fusion. For convenience, we name them GBDT1 (Parameter: tree numbers=800; depth of tree=6; shrinkage=0.05; sampling ratio=1.0; feature ratio=0.6), GBDT2 (Parameter: tree numbers=1000; depth of tree=6; shrinkage=0.05; sampling ratio=1.0; feature ratio=0.6), GBDT3 (Parameter: tree numbers=1000; depth of tree=6; shrinkage=0.05; sampling ratio=0.8; feature ratio=0.8).

TABLE IV. SINGLE MODEL SCORE

Method	**Offline Score**	**Online Score**
LR	7.00%	-
RF	7.30%	-
GBDT1	7.89%	8.66%
GBDT2	7.90%	-
GBDT3	7.91%	-

The results of single GBDT model can have higher score. By combining those results, the performance can be improved further. In the following section, we will introduce model fusion.

VII. Fusion of Multi-Classifier

Fusing the results from different classifier can boost the performance, which has been common knowledge, so, four fusion strategies are tested: ensemble averaging, intersection, LR mixture and stacking GBDT. The **ensemble averaging** method is to sum up the probability from all classifiers as the final output. The **intersection** method is to get the intersection of high positive-probabilistic samples from all classifiers as the output. The **LR mixture** method is to use all probability output of classifiers as the input of LR classifier to learn the weight of each classifier. Stacking GBDT is to train several GBDT classifiers based on different kinds of features and append the features which are the probability output of those GBDTs to the existing features. Based on the new features, a new GBDT classifier is trained as the stacking model.

TABLE V. SCORE OF CLASSIFIER FUSION

Method	**Offline Score**	**Online Score**
Ensemble Averaging: LR+GBDT	7.60%	-

Ensemble Averaging: RF+GBDT	7.72%	-
Ensemble Averaging: LR+RF+GBDT	7.63%	-
Ensemble Averaging: GBDT+GBDT	7.94%	8.70%
Intersection method: LR+GBDT	7.35%	-
Intersection method: RF+GBDT	7.65%	-
Intersection method: LR+RF+GBDT	7.40%	-
Intersection method: GBDT+GBDT	7.92%	-
LR mixture: LR+GBDT	7.83%	-
LR mixture: RF+GBDT	7.81%	-
LR mixture: LR+RF+GBDT	7.85%	-
LR mixture: GBDT+GBDT	7.91%	-
Stacking GBDT	7.95%	8.71%
Ensemble Averaging: Stacking GBDT +GBDT	8.01%	8.78%

Table V shows the performance of different fusion strategies. As the performance of LR and RF is lower than that of GBDT, LR and RF will have negative influence on the final prediction score. Only the combination of several GBDT models outperforms the single GBDT model. The experiments have also shown that the ensemble averaging outperforms intersection and LR mixtures. The best result is generated by summing up all the probability output of the single GBDT model and the stacking GBDT model, which is as the final submission.

VIII. CONCLUSION

In this paper, we present our technical roadmap for AMRA competition organized by Alibaba Group. Our general strategy is that we select some suitable algorithms after comparison and improve the performance of single algorithm as possible by effective training, finally, fuse the result from each algorithm so as to achieve the better performance. Based on this strategy, we design the following technical scheme: First, recommendation task is modeled as the binary classification problem; Then, the hierarchical model is designed; Third, after analyzing the samples, we adopt sample filtering strategy so as to remove the invaluable samples and just remain the valuable samples for training, which can improve the performance of single classifier and speed up the training; Finally, some different classifiers and fusing strategy are tried, and ensemble averaging based on the stacking GBDT is as the final one.

From the aspect of technology, the final results indicate that GBDT seems the better classifier to such classifying problem, and ensemble averaging is the better fusing strategy. From the aspect of competition, our strategy is successful.

Although the competition is ended and our score seems better, we still have a lot of issues to be researched further. For example, deep learning has achieved extremely excellent performance in the fields of computer vision, speech recognition, natural language processing and so on, but we only try it in season 1 not in season 2, therefore, how to apply deep learning to this task is still a challenge; Besides, the final classifier and fusing strategy are both based on GBDT, how to fuse the results of other classifiers effectively is also an open problem.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments and suggestions. This work was supported in part by the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions, by the 111 Project (No.B13022), and by Six talent peaks project in Jiangsu Province.

REFERENCES

[1] Bobadilla, Jesús, et al. "Recommender systems survey." Knowledge-Based Systems 46 (2013): 109-132.

[2] Trevisiol, Michele, et al. "Cold-start news recommendation with domain-dependent browse graph." Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014.

[3] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." Advances in artificial intelligence 2009 (2009): 4.

[4] Gunawardana, Asela, and Christopher Meek. "A unified approach to building hybrid recommender systems." Proceedings of the third ACM conference on Recommender systems. ACM, 2009.

[5] Gunawardana, Asela, and Guy Shani. "A survey of accuracy evaluation metrics of recommendation tasks." The Journal of Machine Learning Research 10 (2009): 2935-2962.

[6] Chen, Xue-Wen, and Xiaotong Lin, "Big data deep learning: Challenges and perspectives." Access, IEEE 2 (2014): 514-525.

[7] Katal, Avita, Mohammad Wazid, and R. H. Goudar. "Big data: Issues, challenges, tools and Good practices." Contemporary Computing (IC3), 2013 Sixth International Conference on. IEEE, 2013.

[8] Wu, Xindong, et al. "Data mining with big data." Knowledge and Data Engineering, IEEE Transactions on 26.1 (2014): 97-107.

[9] Hu, Han, et al. "Toward scalable systems for big data analytics: A technology tutorial." Access, IEEE 2 (2014): 652-687.

[10] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.

[11] Koren, Yehuda. "The bellkor solution to the netflix grand prize." Netflix prize documentation 81 (2009).

[12] Bell, Robert M., and Yehuda Koren. "Lessons from the Netflix prize challenge." ACM SIGKDD Explorations Newsletter 9.2 (2007): 75-79.

[13] Walter, Frank Edward, Stefano Battiston, and Frank Schweitzer. "A model of a trust-based recommendation system on a social network." Autonomous Agents and Multi-Agent Systems 16.1 (2008): 57-74.

[14] McDonald, David W., and Mark S. Ackerman. "Expertise recommender: a flexible recommendation system and architecture." Proceedings of the 2000 ACM conference on Computer supported cooperative work. ACM, 2000.

[15] Breese, John S., David Heckerman, and Carl Kadie. "Empirical analysis of predictive algorithms for collaborative filtering." Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1998.

[16] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." Internet Computing, IEEE 7.1 (2003): 76-80.

[17] Hosmer Jr, David W., and Stanley Lemeshow. Applied logistic regression. John Wiley & Sons, 2004.

[18] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.

[19] Zhou, Tao, et al. "Solving the apparent diversity-accuracy dilemma of recommender systems." Proceedings of the National Academy of Sciences 107.10 (2010): 4511-4515.

[20] X Li, S Qian, et al. "Deep Convolutional Neural Network and Multi-View Stacking Ensemble in Ali Mobile RecommendationAlgorithm Competition." Data Mining Workshop (ICDMW), 2015 International Conference on IEEE, 2015.