Mobile Advertising Predicted Conversion Rate Model
a Recommendation System with Machine Learing Approach

Jiang. Yinghao
Computer Science and Engineering
Sourth University of Science and Technology of China
Shenzhen,Guangdong,China
jiangyh@mail.sustc.edu.cn

Yue. Zhixiong
Computer Science and Engineering
Sourth University of Science and Technology of China
Shenzhen,Guangdong,China
yuezx@mail.sustc.edu.cn

***Abstract*—With the development of mobile internet technology, there have a enormous pontential in mobile advertisement. But how we use this recources becomes a big problem. Fortinately, we can use a recommendation to recommend the advertise for the peolpe the may like it. This is called a excatly advertising putting. It can help the people to get the really information they want, also it can cut down the cost of the company, the can get the consumer approach from which platform, and pay for the cost in datadflow. And the collected data can help the company analysis the user's distribution so that the can improve the production and advertising. So Exactly advertising is one of the most important thing, the effect of advertising, usually measure by clicking and conversion rate in each link, most advertising system by advertising effect data return as the delivery efficiency measure standard to carry out optimization through exposure or click. But how we can trace the user behavior and predicted the advertisement conversion rate. Tecent use the pCVR(Predicted Conversion Rate), to help advertisers tracking advertising.This topic based on the mobile App advertising as the research object, to predict the probability of App ad Click after the activated which is a given advertising, the user and the context condition of advertising is the probability of click after activation. We will try to use KNN, random forest,User-Based top-N recommendation,Time Series model to set up a predict model and verification it in the last for this problem.**

***Keywords*-mobile advertisement; Predicted Conversion Rate; recommendation ; Machine learing;**

I. Introduction

With the development of mobile Internet and communication technology, the number of intelligent terminals and the number of mobile Internet users are explosive growth. It changed the people's lifestyle and business. Individuals begin to get used to shopping, entertainment, information acquisition and other requirements through mobile phones, thus increasing their dependence on mobile APP. The enterprises began to research and develop APP and push it to the market for the purpose of operation, resulting in the promotion demand for APP. Mobile Internet advertising in the rapid rise of APP, but also for the promotion of APP provides an effective marketing approach. In this article we will focus on how to effective putting and how forecast the profit of a advertisement. We will discuss this problem following

II. Data

The data collect and clean by Tencent. The data can be divide into four parts.

A. Advertising features

advertiserID	This is a accountID for a company who buy the advertisement
campaignID	The subsection of advertiserID
adID	The subsection of campaignID
creativeID	The subsection of adID and it is also the content that the user could see
AppID	Which App will be recommend by this creativeID
AppCategory	3 figures the first one represent the first category, the second and third one represent the second category
appPlatform	Android,ios and the other

B. User features

userID	identify the user
age	range from 0 to 80 , 0 for unknown
gender	male,female,unknown
education	The highest education,doesn't distinguish graduate or in school domain primary school , middle school, senior school, bachelor, master, phd
marriage status	single, marriage,unknown
haveBaby	Pregnant,0 6 month, 6 12 month, 1 2year,2 3 year, unknown
hometown	four figures,first and second one represent province and the rest respresent city
appInstallList	until a certain time the application that the user installed filter the high and low frequency app
App install behavior	user install the application in a priod time include install time and appCategory filter the high and low frequency app

C. Context features

positionID	the advtertisement position
sitesetID	the platform for advertisement
positionType	for some site, it has different position type
connectionType	the moblie phone connect to the Internet by which protocal, include 2G,3G,4G,WIFI,unknown
telecoms-Operator	the network service provide by China Mobile, China Unicom, China Telecom or unknown

D. Train Test data

instanceID	to identify the event
lable	the state of the event,not install,installed,to predict
click time	time to click the advertisement
creative ID	see as before
user ID	see as before
positionID	see as before
connection-Type	see as before
telecoms-Operator	see as before

III. Model

A. simple try

1) method: We want to use neural network to solve the problem. Because this problem can be abstracted as a mulitiple domain train and predict the result. so we use follow the diagram to be the input of the neural network training. these are essentially simple mathematical models defining a function function $f : X \to Y$or a distribution over X or both X and Y , but sometimes models are also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase ”ANN model” is really the definition of a class of such functions (where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity).

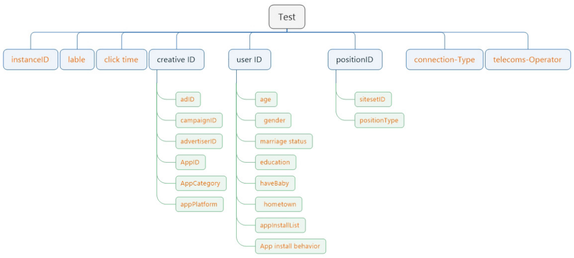


Figure 1. Data structer

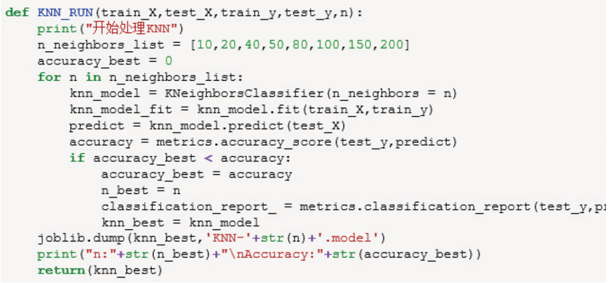
2) data diagram:

3) result: It takes a lot of time to train the model, but the result is not good forus I analysis the result, and think it come out the result because the label is 0 or 1 , and it will be a time dealy. But the neural network doesn’t consider the time delay,and time will be a important element in this prediction. And we should use some model to separate the advertisement and personal characteristics. We learn from the recommand system and set up a model.

B. Advertising data model

1) method: We can choose a variable $Q_{ad}$to determine the advertisement quility. And using clustering method to classification the advertisement. The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields.
The k-means algorithm divides a set of N samples X into K disjoint clusters C, each described by the mean μ of the samples in the cluster. The means are commonly called the cluster centroids; note that they are not, in general, points from X, although they live in the same space. The K-means algorithm aims to choose centroids that minimise the inertia, or within-cluster sum of squared criterion:

$$\sum_{i=0}^{n} min(||x_j - \mu_i||^2)$$



```
def KNN_RUN(train_X,test_X,train_y,test_y,n):
    print("开始处理KNN")
    n_neighbors_list = [10,20,40,50,80,100,150,200]
    accuracy_best = 0
    for n in n_neighbors_list:
        knn_model = KNeighborsClassifier(n_neighbors = n)
        knn_model_fit = knn_model.fit(train_X,train_y)
        predict = knn_model.predict(test_X)
        accuracy = metrics.accuracy_score(test_y,predict)
        if accuracy_best < accuracy:
            accuracy_best = accuracy
            n_best = n
            classification_report_ = metrics.classification_report(test_y,p
            knn_best = knn_model
    joblib.dump(knn_best,'KNN-'+str(n)+'.model')
    print("n:"+str(n_best)+"\nAccuracy:"+str(accuracy_best))
    return(knn_best)
```

Figure 2. advertising culstering code

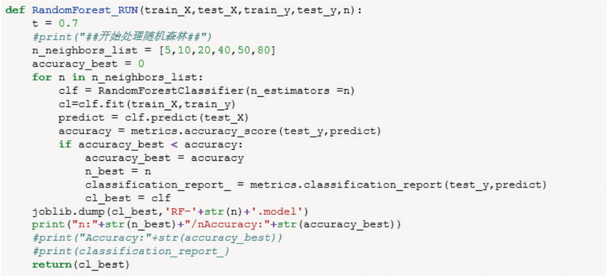
2) code:

C. User feature model

1) random forest: Becaue the question can easily saw as user to choose wheather the install the app, it likes the random forst.Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees’ habit of overfitting to their training set.
Given a training set $X = x_1, ..., x_n$ with reponses $Y = y_1, ..., y_n$ bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples: for $b = 1, ..., b$
(1)Sample, with replacement, B training examples from

X, Y; call these x_b, Y_b.
(2)Train a decision or regression tree f_b on X_b, Y_b.
After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$



```
def RandomForest_RUN(train_X,test_X,train_y,test_y,n):
    t = 0.7
    #print("##开始处理随机森林##")
    n_neighbors_list = [5,10,20,40,50,80]
    accuracy_best = 0
    for n in n_neighbors_list:
        clf = RandomForestClassifier(n_estimators =n)
        cl=clf.fit(train_X,train_y)
        predict = clf.predict(test_X)
        accuracy = metrics.accuracy_score(test_y,predict)
        if accuracy_best < accuracy:
            accuracy_best = accuracy
            n_best = n
            classification_report_ = metrics.classification_report(test_y,predict)
            cl_best = clf
    joblib.dump(cl_best,'RF-'+str(n)+'.model')
    print("n:"+str(n_best)+"/nAccuracy:"+str(accuracy_best))
    #print("Accuracy:"+str(accuracy_best))
    #print(classification_report_)
    return(cl_best)
```

Figure 3. random forest code

2) User-Based top-N recommendation: Beacuse we have many advertisement to recommendation, and have a large number of user information, we can use the user-based top-N recommendation algorithm. Particular classes of model-based top-N recommendation algorithms that build the recommendation model by analyzing the similarities between the various items and then use these similar items to identify the set of items to be recommended.
Define symbols n and m to denote the number of distinct users and the number of distinct items in a particular dataset, respectively. And here we will use the symbol N to denote the number of recommendations that needs to be computed for a particular user.Represent each dataset by an n m binary matrix R that will be referred to as the user-item matrix, such that R(i,j) is one if the i-th customer has purchased the j-th item, and zero otherwise. The main idea of the algorithm is as below:Given a useritem matrix R and a set of items U that have been purchased by a user, identify an ordered set of items X such that $|x| \leq N$ and $X \bigcap U = \emptyset$

D. Time Series

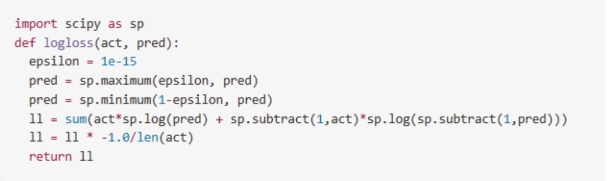
But all of above didn't include the time influence, because it is provide the data that in a right time. So we need the time series model to predict the the label value. Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values. While regression analysis is often employed in such a way as to test theories that the current values of one or more independent time series affect the current value of another time series, this type of analysis of time series is not called "time series analysis", which focuses on comparing values of a single time series or multiple dependent time series at different points in time.

IV. Result

I reserve the last day data to check the model. Use the Logarithmic Loss:

$$logloss = -\frac{1}{N} \sum_{i=1}^{N} (y_i log(p_i) + (1 - y_i) log(1 - p_i))$$

N is the amount of the test data,y_i value of 0 or 1. p_i is the probility of the label become 1.



```
import scipy as sp
def logloss(act, pred):
  epsilon = 1e-15
  pred = sp.maximum(epsilon, pred)
  pred = sp.minimum(1-epsilon, pred)
  ll = sum(act*sp.log(pred) + sp.subtract(1,act)*sp.log(sp.subtract(1,pred)))
  ll = ll * -1.0/len(act)
  return ll
```

Figure 4. check code

Acknowledgments

This work was partially supported by SUSTech fund (05/Y01051814, 05/Y01051827, 05/Y01051830, 05/Y01051839).

References

[1] Kim H W,Lee H L,Son J E. An Exploratory Study on the Determinant of Smartphone App Purchase[J].The journal of Society for E-Business Studies,2011,16(4):173-196

[2] Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". Psychological Review. 65 (6): 386408. PMID13602029. doi:10.1037/h0042519.

[3] Damas, M.; Salmeron, M.; Diaz, A.; Ortega, J.; Prieto, A.; Olivares, G. (2000). "Genetic algorithms and neuro-dynamic programming: application to water supply networks". Proceedings of 2000 Congress on Evolutionary Computation. 2000 Congress on Evolutionary Computation. La Jolla, California, USA: IEEE. ISBN0-7803-6375-2. doi:10.1109/CEC.2000.870269. Retrieved 29 July 2012.

[4] Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. 46 (3): 175185. doi:10.1080/ 00031305.1992.10475879

[5] Ho, Tin Kam (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832844. doi:10.1109/34.709601

[6] Kleinberg, Eugene (1996). "An Overtraining-Resistant Stochastic Modeling Method for Pattern Recognition" . Annals of Statistics. 24 (6): 23192349.

[7] By B Sarwar, G Karypis, J Konstan, J Riedl. Proceedings of the 2nd ACM conference on Electronic commerce, 2000:158-167

[8] Nikoli, D.; Muresan, R. C.; Feng, W.; Singer, W. (2012). "Scaled correlation analysis: a better way to compute a cross-correlogram". European Journal of Neuroscience. 35 (5): 742762. doi:10.1111/j.1460-9568.2011.07987.x.