ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 1276 -1282
doi: 10.18178/wcse.2017.06.222

Enhanced Online Taxi Orders Prediction Based on User Behavior

Ningyuan Huang, Jiaxing Song [+], Weidong Liu

Department of Computer Science and Technology, Tsinghua University, Beijing, China

Abstract. With the development of online taxi companies such as Uber and UCAR, it is not difficult to connect passengers and taxis for hire anymore. Indeed, these companies are concerning about predicting distribution of taxi requests and dispatching their cars properly nowadays. In this paper, the algorithm addressed focuses on discovering user behaviors to enhance traditional prediction algorithm. Through the experiment, the algorithm is testified and achieves a good result on real data.

Keywords: classification, taxi order prediction, spatial data mining

1. Introduction

With the continuous improvement of computer technology and the popularity of the Internet, online taxi services blowout recent years. In the past, taxi drivers could only use experiences to guess where the requests was. Since the e-hailing Apps are spreading, drivers and taxi companies can not only know where the requesting client exactly is, but also get information where the vehicles are inadequate.

When the passengers need taxis and cars for hire are far away from the requesters, it will be poor experiences since the companies need to move their vehicles from distant areas. If we can predict when and where the requests will appear, the companies will redistribute their drivers to fit the coming requests.

So the problem of today's era becomes predicting passengers' requests and dispatching drivers appropriately to enhance user experience.

In this paper, we introduce an algorithm Behavior Relation Prediction (BRP). Unlike other methods, BRP not only concerns where and when the orders happens, but also focuses which client creates the order. First, we split orders into two parts: one contains frequent behaviors of the clients and the other contains the remaining orders. Then, we use BRP to predict the frequent behaviors while we use an existing method to predict the others. Finally, we add the two part together and give our final prediction of order distribution in the next period of time.

The organization of the rest of the paper is as follows. In Section 2, some works about taxi are mentioned first and introduction of some significant algorithms comes next. In Section 3, some terms and the definition of problem are listed. In Section 4, the approach and the algorithm of the problem is introduced. In section 5, the algorithm is implemented to a real dataset and compared with some traditional algorithms. Finally, we conclude with a summary in Section 6.

2. Related Work

Previous works, such as [1–3], focus on offline taxi service and use clustering algorithms to find out where the hot spots are in order to lead both drivers and passengers finding each other. However it cannot fit our operators' needs today. Tian in [4] use Gauss Mixture Experts to predict the proportion of blocks to

[+] Corresponding author. Tel.: +8613911567550; fax: +861062781446.
E-mail address: jxsong@tsinghua.edu.cn.

predict taxi orders, which gives good results in some hot areas and specific time spans. None of them use information of individual clients to find out their laws to call a taxi.

It is a simple thought using a classifier to predict whether a behavior will happen soon. Classifying algorithm has been developing for decades, SVM (Support Vector Machine), Decision Tree are the most common classifier these days. C. Cortes and V. Vapnik [5] introduce non-linear kernel into SVM to adapt high dimensional features. Hence, it is widely used to in the two-group classification problems. Decision Tree Classifier [7] is also a fundamental approach in classification.

Agrawal [8] introduces a method mining association between sets of items. His work is quite significant about the criteria of association rules.

3. Problem Definition

For the companies, the urban area is split into several blocks like Fig. 1 and they want to know the number of orders in the next period of time in every block of the city. Here are some terms we will use in this paper:

Table 1: Variables and Functions of the Problem

Variable or Function	Description
c	Client ID
C	Set of all client IDs
t	Estimate board time
p	Estimate board position, consists of latitude and longitude
b_i	Block in the urban area where we want to predict. $\forall_{i \neq j} b_i \cap b_j = \emptyset$
$beh_{c,i}$	Behavior of client c that describes his order in block $b_{c,i}$ at hour $h_{c,i}$
$o(c, p, t)$	If client c has a request to board at position p, time t, then $o(c, p, t) = 1$, otherwise it is assigned 0
$o(p, t)$	Number of total orders at time t, position p
$o(c,b_i,\tau)$	Number of total orders that client c requests in block b_i in the period τ
$\eta(c,b_i,\tau)$	Expectation number of total orders that client c requests in block b_i in the period τ
$o(b_i,\tau)$	Number of total orders in block b_i in the period τ
$\eta(b_i,\tau)$	Expectation number of total orders in block b_i in the period τ
$weekday(t)$	Weekday of the time t, 1 represents Monday, 7 represents Sunday
$hour(t)$	Hour of the time t

Fig. 1: Split area into blocks.

Our target is to find a function $\eta(p,t) = \sum_{c \in \mathbf{C}} o(c,p,t)$ that $\eta(b_i,\tau) = \int_{p \in b_i, t \in \tau} \eta(p,t) dpdt$ is the number of expected orders in b_i during period τ.

4. Approach

4.1. Problem Reduction

On the one hand, if one block has a huge size, even we can precisely predict the number of orders and there are adequate cars for hire locating in the given block, it can take quite long time for drivers to pick up their passengers in the same block, on the other hand, if a block is extremely small, lack of data will lead to overfitting problem. Likely, long predicting period causes resource wasting since drivers tends to wait for a long time and short period results in overfitting. Therefore, the size of blocks and predicted period should be assigned properly.

To simplify the problem, we just split time into unit of hour and use Geohash algorithm to separate the blocks. The precision of Geohash parameter is set by 6 characters, which UCAR is using on their prediction system.

4.2. User Behavior Analyzing

Fig.2 shows the order count of two weeks in a block. Lots of people schedule their work or life on a weekly basis. Predicting these behavior is quite simple and easy. But there are still some rules we can find besides weekly routine. For example, a professor is attending a conference and goes to conference hall from hotel every morning. In another scene, an athlete goes to a training gym frequently but irregularly and takes round-trip taxi every time. Obviously, they are not a weekly schedule. But we can use different ways to predict behaviors like them.

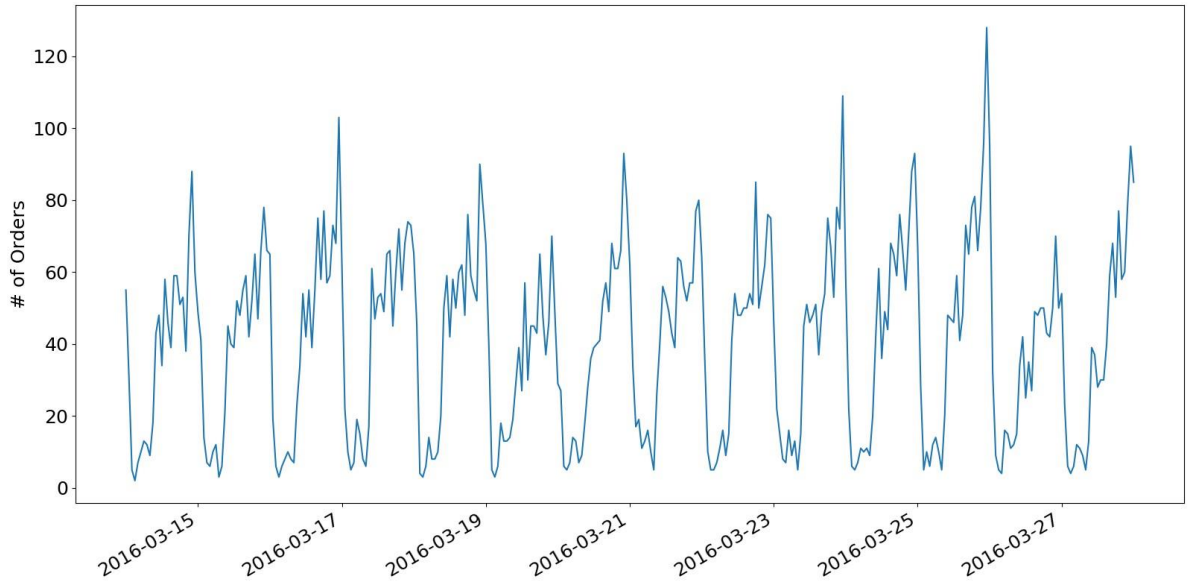

Fig. 2: Order count by hour.

The instinct to find the relationship between behaviors is using association rule mining algorithms. But it is unwise to build “transactions” by splitting time at midnights in case the following behavior happens in the next day of its precedent. However, the criteria of mining association rules is enlightening.

4.3. Behaviors to Predict

Since we use Geohash to split area into blocks and split time by hour, it is simple to classify orders into the clusters by boarding positions and hour of estimate boarding time. For every passenger, a Geohash code and a hour number describe a behavior. For example, if an order’s board position is 116.506622°*E*, 40.016418°*N* and board time is 2016-04-13 13:54:10, we use *wx4geh 13* to describe it. If a behavior appears more than once, we guess it will be predictable. We want to find out in which days the behavior will happen.

So the orders in predictable behaviors are extracted to find out the rules while the others are to be predicted in a simpler method.

4.4. Predicting Behaviors

- Building Training Set

For each behavior $beh_{c,i} = \{\exists p,t | o(c,p,t) = 1, p \in b_{c,i}, hour(t) = h_{c,i}\}$ of client c to be predicted, we can collect the client's all predictable behaviors in the past few weeks. The input to the training algorithm is a set of examples $\vec{x_k}$ and labels y_k.

Assume client c has n behaviors $\{beh_{c,1}, beh_{c,2}, ..., beh_{c,n}\}$ and the orders in $beh_{c,i}$ happened in hour $h_{c,i}$ in their days. Then for each day k,

$$x_{kl} = \begin{cases} 1, \exists p,t, p \in b_{c,l} \text{ and } 0 < time(day\ k, hour\ h_{c,i}) - t \le 24\ hours \\ 0, others \end{cases}, \ 1 \le l \le n$$

$$x_{kl} = \begin{cases} 1, weekday(k) = l - n \\ 0, others \end{cases}, \ n+1 \le l \le n+7$$

$$x_{kl} = weekday(k), \ l = n+8$$

$$y_k = \begin{cases} 1, \exists p,t, p \in b_{c,i}, hour(t) = h_{c,i}, o(c,p,t) = 1 \\ 0, others \end{cases}$$

Label y_k describes whether the predicting behavior happened in day k, hour h. In $\vec{x_k}$, the first n component describe whether the behaviors happened in the past day of the given time, while the last components describes the day k.

- Evaluation Function

After predicting the behaviors, we add the predictions together to get the expectation of orders in the given block b_i and given hour τ. Then

$$\eta(b_i, \tau) = \sum_{c \in \mathbf{C}} \eta(c, b_i, \tau)$$

$$o(b_i, \tau) = \sum_{c \in \mathbf{C}} o(c, b_i, \tau)$$

We use *Absolute Error* (AE) as our evaluation function. So our target is to minimize all $AE(b_i,\tau) = |\eta(b_i,\tau) - o(b_i,\tau)|$.

- Relative Behavior Prediction

As mentioned above, we want to find out whether each behavior happens with a weekly rule or is caused by a precedent behavior. Thus we need to calculate the probability of each case. We assume that the given client has n predictable behaviors and $beh_{c,i}$ is the one to be predicted, X is the attribute matrix consists the set of $\vec{x_k}$, Y consists of corresponding y_k, then we use a similar way to association rule mining to calculate the probabilities of precedent behavior and weekly behavior:

```
PROB_OF_REL(X, Y, n)
1  for i ← 1 to n
2      do correct ← 0
3         for j ← 1 to length[Y]
4             do if X[j][i] = Y[j]  ▷ 2 behaviors both happened or both not happened
5                   then correct ← correct + 1
6         prob[i] = correct/ length[Y]
7  Return prob
```

The probability of associate precedent behavior is the precision if we use the behavior to predict $beh_{c,i}$ in the training set, and the probability of weekly behavior represents the precision if we predict $beh_{c,i}$ by the majority of the same weekday.

```
PROB_OF_WEEK(X, Y, n)
 1  weeks = length[Y]/7
 2  weekday_count = zeros
 3  for i ← 1 to length[Y]
 4       do if Y[i] = 1
 5            then weekday_count[X[i][n + 8]] ← weekday_count[X[i][n + 8]] + 1
 6  correct ← 0
 7  for i ← 1 to length[Y]
 8       do if weekday_count[i] > weeks/2
 9            then correct ← correct + weekday_count[i]
10            else correct ← correct + weeks − weekday_count[i]
11  prob_week ← correct/ length[Y]
12  Return prob_week
```

Then we can examine our model using the testing set. After using the same way to build $\vec{x}$, we fill the weight vector $\vec{w}$ with the rule follows: if the largest probability belongs to behavior $beh_{c,j}$, then w_j is assigned with 1; on the other hand, the probability of weekly schedule dominates, the corresponding component should be assigned with the average value in the weekday of the training set. Finally, the prediction value is $\vec{x} \cdot \vec{w}$.

Since the behavior itself also appears in matrix X, it is easy to prove that a new daily behavior will be predicted to happen in the third day using the algorithm, which solves the 'professor' case mentioned in Section 3.

5. Experiment

The historical orders records are the basic components of the system, and the quality of dataset influences the performance as well.

5.1. Dataset

The dataset is based on UCAR's data of Beijing in March and April, 2016. 1,942,269 orders from 342,331 clients are imported. In the 2 months, we find out behaviors to predict, 257,410 orders from 10,998 different clients are extracted.

We use data from April 1st to April 7th as testing set and use data in the past 3 weeks of testing set to predict. There are 31,607 orders to predict.

5.2. Behavior Prediction Implementation

- 5.2.1 Baseline

There are several classifying algorithm developed. In this work, we use *SVM* (Support Vector Machine) and *Decision Tree* as baseline of behavior prediction. Further, we use *Weekly Mean* to set a reference of baseline where

$$\eta_{WM}(b_i, \tau) = \frac{\sum_{j=1}^{k} o(b_i, \tau - j * (1\ week))}{k}$$

In this work, $k = 3$ is assigned.

Finally, we use the AE/Actual ratio $\sum AE\ (b_i, \tau) / \sum o(b_i, \tau)$ to evaluate the performance of each algorithm.

- 5.2.2 Performance of Algorithms

Table 2 shows the performance of *Behavior Relation Prediction*, *SVM*, *Tree* and *Weekly Mean*. The error of *BRP* is less than half of the others. Fig. 3 shows the Mean Absolute Error in the hours of different classifiers. Obviously, *BRP* dominates the others in most predictions. Since there are so many factor that impact people to plan their weekly schedule, using recent data gives us more information than the data far before. Indeed, we have few records for each behavior, which implies complicate algorithms like SVM or Decision Tree would not take advantage but the simpler method BRP could deal with the problem.

Table 2: AE/Actual Ratio of Different Algorithms

	BRP	SVM	Tree	WM
AE/Actual	0.18994	0.39624	0.39381	0.38503

5.3. Order Distribution Prediction

After predicting behaviors, we add the result back to the full dataset with 167,795 orders. First, we use WM to predict the whole set as a baseline. Then, we use BRP to predict the predictable behaviors ($\eta_{BRP}(b_i,\tau)$) while using WM to predict the others ($\eta_{WM\ (others)}(b_i,\tau)$), that $\eta(b_i,\tau) = \eta_{WM\ (others)}(b_i,\tau) + \eta_{BRP}(b_i,\tau)$. BRP decreases the AE/Actual ratio from 0.872 to 0.847.

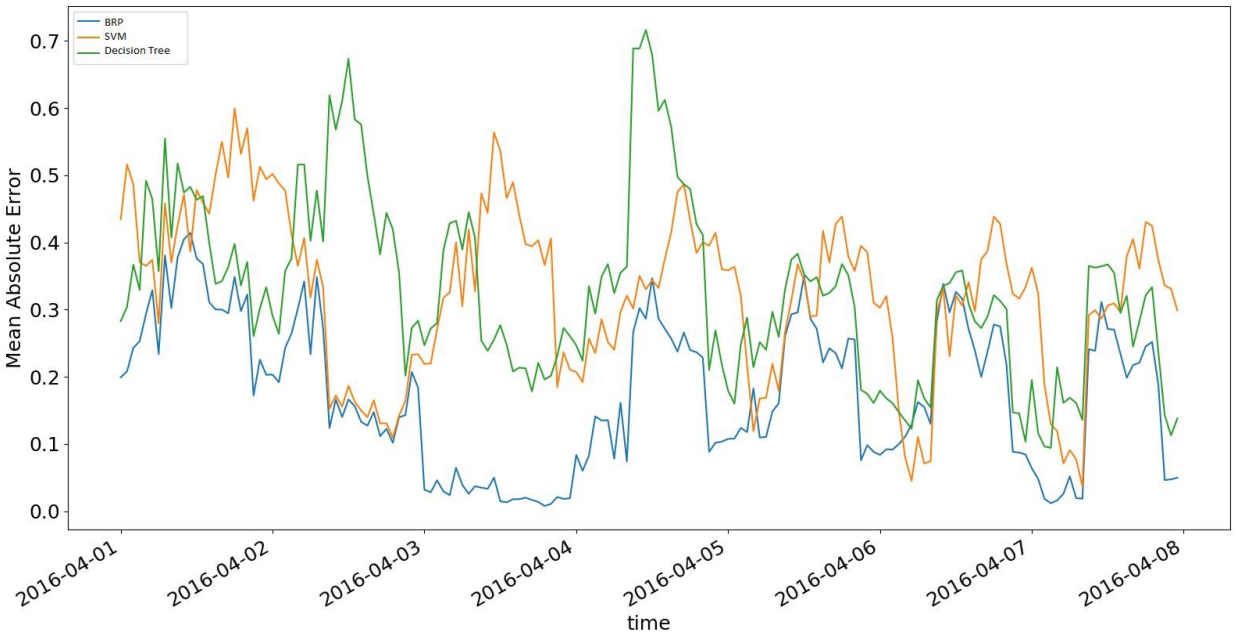

Fig. 3: Mean absolute error against time of different algorithms.

6. Conclusion

In this work, we approach to enhance order distribution prediction of online taxi request by predicting relationship of user behaviors. Finding out the relationship of behaviors, we can find out how and why the orders happened clearly, which leads to a better prediction performance.

Our algorithm is not only good at finding relationship between behaviors, but also responding quickly to the short-term behaviors.

Although the increment of enhancing is not significant, the results of behavior prediction are still useful for the companies. The clients who experience the service regularly are worthy of attention and deserve better performance. Since we can learn exactly where and when the behaviors happen, the companies can move their vehicles in advance to optimize user experience of their service.

7. References

[1] Chang, Han-wen, Yu-chin Tai, and Jane Yung-jen Hsu. ”Context-aware taxi demand hotspots prediction.” International Journal of Business Intelligence and Data Mining 5.1 (2009): 3-18.

[2] Yuan, N. J., Zheng, Y., Zhang, L., Xie, X. (2013). ”T-finder: A recommender system for finding passengers and vacant taxis”. Knowledge and Data Engineering, IEEE Transactions on, 25(10), 2390-2403.

[3] Zheng, X., Liang, X., Xu, K. (2012, August). ”Where to Wait for a Taxi”. In Proceedings of the ACM SIGKDD International Workshop on Urban Computing (pp. 149-156). ACM.

[4] Yong Tian, Ningyuan Huang, Weidong Liu, Jiaxing Song. ”Urban Trip Requests Prediction: An Operators Perspective.” WCSE 2016.

[5] Cortes, Corinna, and Vladimir Vapnik. ”Support-vector networks.” Machine learning 20.3 (1995): 273-297.

[6] Vapnik, Vladimir, Isabel Guyon, and Trevor Hastie. ”Support vector machines.” Machine learning 20 (1995): 273-297.

[7] Safavian, S. Rasoul, and David Landgrebe. ”A survey of decision tree classifier methodology.” IEEE transactions on systems, man, and cybernetics 21.3 (1991): 660-674.

[8] Agrawal, Rakesh, Tomasz Imieliski, and Arun Swami. ”Mining association rules between sets of items in large databases.” Acm sigmod record. Vol. 22. No. 2. ACM, 1993.