A Taxi Business Intelligence System

Yong Ge[1], Chuanren Liu[1], Hui Xiong[1], Jian Chen[2]
[1]Rutgers Business School, Rutgers University
yongge@pegasus.rutgers.edu, {hxiong, chuanren.liu}@rutgers.edu
[2]Tsinghua University
jchen@mail.tsinghua.edu.cn

ABSTRACT

The increasing availability of large-scale location traces creates unprecedent opportunities to change the paradigm for knowledge discovery in transportation systems. A particularly promising area is to extract useful business intelligence, which can be used as guidance for reducing inefficiencies in energy consumption of transportation sectors, improving customer experiences, and increasing business performances. However, extracting business intelligence from location traces is not a trivial task. Conventional data analytic tools are usually not customized for handling large, complex, dynamic, and distributed nature of location traces. To that end, we develop a taxi business intelligence system to explore the massive taxi location traces from different business perspectives with various data mining functions. Since we implement the system using the real-world taxi GPS data, this demonstration will help taxi companies to improve their business performances by understanding the behaviors of both drivers and customers. In addition, several identified technical challenges also motivate data mining people to develop more sophisticate techniques in the future.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms, Experimentation

Keywords

Business Intelligence, Route Recommendation, Taxi Driving Fraud Detection

1. INTRODUCTION

Advances in sensor, wireless communication, and information infrastructures such as GPS, WiFi and RFID have enabled us to collect large amounts of location traces (trajectory data) of individuals or objects. Such a large number of trajectories provide us unprecedented opportunity to automatically discover useful knowledge, which in turn deliver intelligence for real-time decision making in various fields, such as route recommendations and fraud detection.

In this demonstration, we introduce several interesting applications, which are enabled by both the availability of taxi GPS traces and advanced data mining techniques. First, we exploit the knowledge extracted from taxi location traces and develop a route recommendation function based on business success metrics. The key idea is to leverage the business knowledge from the historical data of successful taxi drivers for helping other taxi drivers to improve their business performances. Specifically, this function is able to recommend a driving route, which is essentially a sequence of pick-up points, to taxi drivers. The suggested optimal route is expected to result in the minimal driving distance before the next pick-up event. Since there are usually hundreds of pick-up points in a city and this is naturally a combinatorial problem, efficient algorithms are required to search the optimal route for online recommendation.

In addition, we understand drivers' behaviors from taxi GPS traces and develop a taxi driving fraud detection function. Indeed, fraudulent taxi drivers often commit driving fraud activities and overcharge passengers by deliberately taking unnecessary detours. To detect taxi driving fraud activities, we need to consider different aspects of taxi driving activities, such as traces, speed, time, distance, and traffic situations, which can be revealed from the GPS traces.

Finally, in the demo system, we also provide a function to illustrate some business insights by exploiting the taxi data. The business insight, such as the tip distribution and the economic potential of pick-up points, can guide taxi drivers to improve their performances.

A screenshots of this online demonstration system[1] is shown in Figure 1. We provide a user-friendly interface, which allows user to operate the different functions easily. Also, most output of functions are visualized using the Google Map API. This helps users better explore the results. Finally, two sets of real-world taxi data are collected and explored in the demo system. One is collected from 500 taxi drivers for about 30 days in the San Francisco Bay Area. Another one is collected from 24 taxi drivers in the New York City area for two year period.

2. DRIVING ROUTE RECOMMENDATION

Consider a scenario that a large number of GPS traces of taxi drivers have been collected for a period of time. In this

[1]http://kdd11demo.appspot.com/app.htm

KDD'11, August 21–24, 2011, San Diego, California, USA.

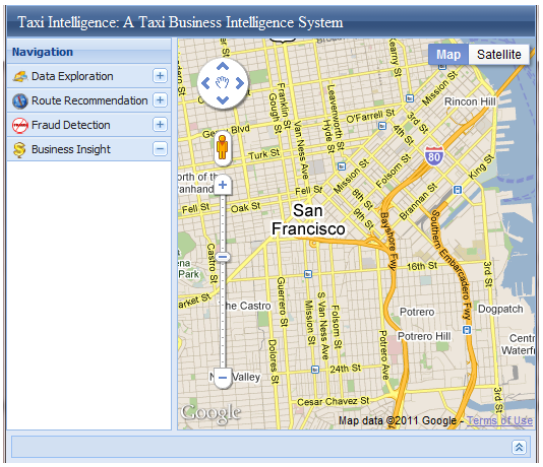


Figure 1: A Screen Shot of the Demo System

collection of location traces, we also have the information about when a cab is available or occupied. In this data set, it is possible to first identify a group of taxi drivers who are very successful in business. Then, we can cluster the pick-up points of these taxi drivers for a certain time period. The centroids of these clusters can be used as the recommended pick-up points with a certain probability of success for new taxi drivers in these areas. Then, a mobile sequential recommendation problem can be described as follows.

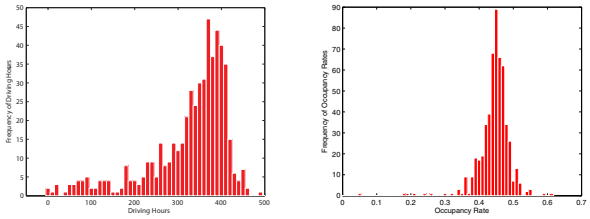
Assume that a set of N potential pick-up points, $\mathcal{C}$=$\{C_1, C_2, \cdots, C_N\}$, is available. Also, the estimated probability that a pick-up event could happen at each pick-up point is known as $P(C_i)$, where $P(C_i)(i = 1, \cdots, N)$ is assumed to be independently distributed. Let $\mathcal{P} = \{P(C_1), P(C_2), \cdots, P(C_N)\}$ denote the probability set. In addition, let $\overrightarrow{\mathcal{R}} = \{\overrightarrow{R_1}, \overrightarrow{R_2}, \cdots, \overrightarrow{R_M}\}$ be the set of all the directed sequences (potential driving routes) generated from $\mathcal{C}$ and $|\overrightarrow{\mathcal{R}}| = M$ is the size of $\overrightarrow{\mathcal{R}}$ - the number of all possible driving routes. Note that the pick-up points in each directed sequence are assumed to be different from each other. Next, let $L_{\overrightarrow{R_i}}$ be the length of route $\overrightarrow{R_i}(1 \leq i \leq M)$, where $1 \leq L_{\overrightarrow{R_i}} \leq N$. Finally, for a directed sequence $\overrightarrow{R_i}$, let $\mathcal{P}_{\overrightarrow{R_i}}$ be the route probability set which includes the probabilities of all pick-up points containing in $\overrightarrow{R_i}$, where $\mathcal{P}_{\overrightarrow{R_i}}$ is a subset of $\mathcal{P}$.

The objective of this MSR problem is to recommend a travel route for a cab driver in a way such that the potential travel distance before having customer is minimized. We defined a function $\mathcal{F}$ to compute the Potential Travel Distance (PTD) before having a customer. The PTD can be denoted as $\mathcal{F}(PoCab, \overrightarrow{\mathcal{R}}, \mathcal{P})$. In other words, the computation of PTD depends on the current position of a cab (PoCab), a suggested sequential pick-up points ($\overrightarrow{\mathcal{R}}$), and the corresponding probabilities associated with all recommended pick-up points.

The MSR problem involves the recommendation of a sequence of pick-up points and has combinatorial complexity in nature. We have developed some pruning techniques [4] to efficiently search the optimal route.

2.1 High-Performance Drivers

In real world, there are always high-performance experienced cab drivers, who typically have sufficient driving hours and higher customer occupancy rates - the percentage of



(a) Driving Hours (b) Occupancy Rates

Figure 2: Some Statistics of the Cab Data.

driving time with customers. For example, Figure 2 (a) and (b) show the distributions of driving hours and occupancy rates of more than 500 drivers in San Francisco over a period of about 30 days. In the figure, we can clearly see that the drivers have different performances in terms of occupancy rates. Based on this observation, we will first extract a group of high-performance drivers with sufficient driving hours and high occupancy rates. The past pick-up records of these selected drivers will be used for the generation of potential pick-up points for recommendation.

2.2 Clustering Based on Driving Distance

After carefully observing historical pick-up points of high-performance drivers, we notice that there are relative more pick-up events in some places than others. In other words, there are the cluster effect of historical pick-up points. Therefore, we propose to cluster historical pick-up points of high-performance drivers into N clusters. The centroids of these clusters will be used for recommending pick-up points. For this clustering algorithm, we use driving distance rather than Euclidean distance as the distance measure. In this study, we perform clustering based on driving distance during different time periods in order to have recommending pick-up pointers for different time periods. Another benefit of clustering historical pick-up points is to dramatically reduce the computational cost of the MRS problem.

2.3 Probability Calculation

For each recommended pick-up point (the centroid of historical pick-up cluster), the probability of a pick-up event can be computed based on historical pick-up data. The idea is to measure how frequent pick-up events can happen when cabs travel across each pick-up cluster. Specifically, we first obtain the spatial coverage of each cluster. Then, let $\#_T$ denote the number of cabs which have no customer before passing a cluster. For these $\#_T$ empty cabs, the number of pick-up events $\#_P$ is counted in this cluster. Finally, the probability of pick-up event for each cluster (each recommended pick-up point) can be estimated as $P(C_i)_{1 \leq i \leq N} = \frac{\#_P}{\#_T}$, where $\#_P$ and $\#_T$ are recorded for each historical pick-up cluster at different time periods.

2.4 The Recommendation Process

Even though we can find the optimal drive route for a given cab with its current position, it is still a challenging problem about how to make the recommendation for many cabs in the same area. In this section, we address this problem and introduce a strategy for the recommendation process in the real world.

A simple way is to suggest all these empty cabs to follow the same optimal drive route, however there is naturally an

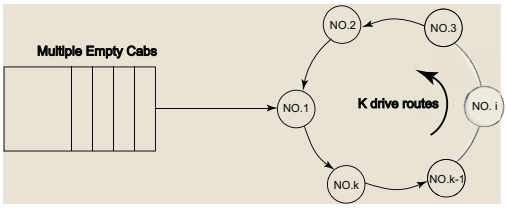


Figure 3: Illustration of the *Circulating Mechanism*.

overload problem, which will degrade the performance of the recommender system. To this end, we employ load balancing techniques [6] to distribute the empty cabs to follow multiple optimal drive routes. The problem of load balancing has been widely used in distributed systems for the purpose of optimizing a given objective through finding allocations of multiple jobs to different computers. For example, the load balancing mechanism distributes requests among web servers in order to minimize the execution time. For the proposed mobile recommendation system, we can treat multiple empty cabs as jobs and multiple optimal drive routes as computers. Then, we can deal with this overload problem by exploiting existing load balancing algorithms. Specifically, in this study, we apply the *circulating mechanism* for the recommender systems by exploiting a Round Robin algorithm [8], which is a static load balancing method.

Under the *circulating mechanism*, to make recommendation for multiple empty cabs, a round robin scheduler alternates the recommendation among multiple optimal drive routes in a circular manner. As shown in Figure 3, we could search k optimal drive routes and recommend the NO.1 route to the first coming empty cab. Then, for the second empty cab, the NO. 2 drive route will be recommended. Assume there are more than k empty cabs, recommendations are repeated from NO. 1 route again after the kth empty cab. In practice, to achieve this, one central dispatch (processor) is needed to maintain the empty cabs and assignments among the top-k driving routes. Note that the load balancing techniques are not the focus of this paper.

3. DRIVING FRAUD DETECTION

Taxi driving frauds are often committed by greedy taxi drivers who overcharge passengers by deliberately taking unnecessary detours. Nowadays, many taxi service complains are related to taxi driving frauds [2, 3, 1]. Therefore, it becomes invaluable for improving taxi services by providing the information about taxi driving frauds. However, it is a challenging issue to detect driving fraud activities committed by experienced and cunning taxi drivers who know how to manipulate the driving routes to commit driving frauds without being disclosed by passengers.

A promising direction to solve this problem is to collect and analyze the GPS traces by taxi drivers. Indeed, GPS tracking devices have been installed in many city taxis and a large amount of GPS traces has been accumulated for the analysis. These GPS traces provide unparallel opportunities for us to develop new ways to uncover taxi driving fraud activities. To that end, in this demonstration, we develop a taxi driving fraud detection function by exploiting GPS traces by taxi drivers.

Even there are already various anomaly detection techniques [5, 7], we are still facing several essential challenges in order to develop a reliable taxi driving fraud detection system. For example, we need to deal with different sets of

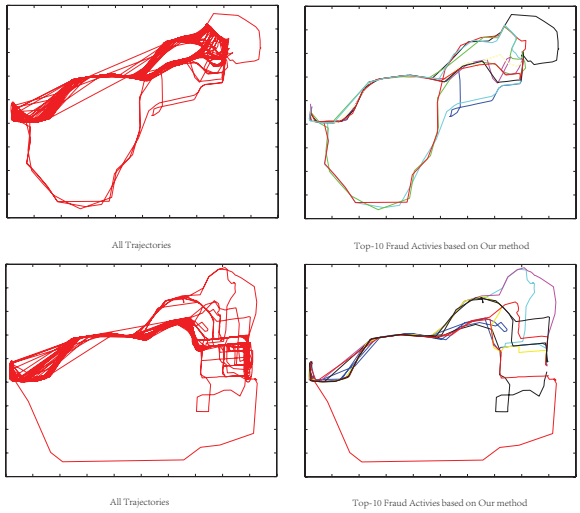


Figure 4: Driving Fraud Activities

trajectories between different pairs of source and end nodes. These trajectories are heterogeneous in nature. Also, we need to distinguish some shortcuts, which are abnormal, from real frauds. In addition, a practical challenge is that some drivers, who commit a driving fraud, may be truly unfamiliar with the local region or may use this as an excuse. Similarly, drivers may argue that they detour due to the traffic situations.

To deal with these practical challenges, we provide different approaches to find two aspects of evidences: travel route evidences and driving distance evidences. Furthermore, these two aspects of evidences are well combined together by using Dempster-Shafer theory. In addition, various customized and suitable techniques are applied to find each evidence. To perform taxi fraud detection, we first identify each pair of source and end nodes, and the trajectories from source to end nodes. For example, in Figure 4, we show some taxi driving fraud activities detected by our approach for two pairs of source and end nodes.

3.1 Regularity of Fraud Activities

After detecting the fraud activities for all pairs of source and end nodes, we are able to count the number of driving fraud activities for each driver. In Figure 5 (a), we show the frequency of driving fraud activities for each driver within about 30 days. We highlight top-3 drivers, who commit the most frauds. To further observe if some drivers habitually commit driving fraud every day, we show the fraud activities day by day for each driver in Figure 5 (b), where each dot represents one fraud activity by a driver in one day. As can be seen, some drivers do have relatively habitual fraud behaviors, such as those top-3 drivers.

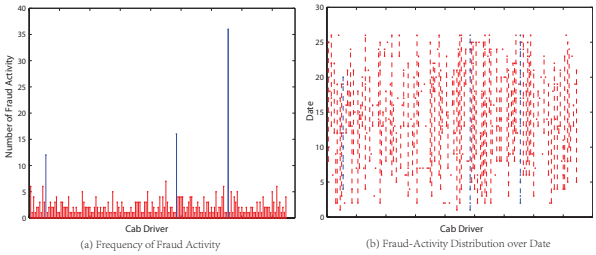


Figure 5: Illustration of Fraud Activities.

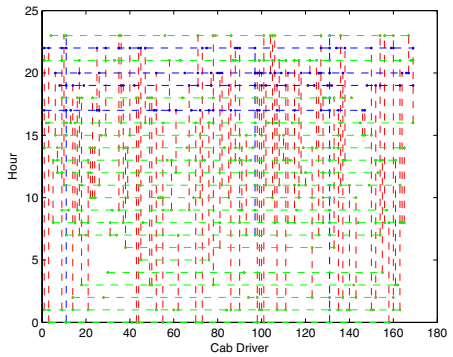


Figure 6: Temporal Distribution of Fraud Activities.

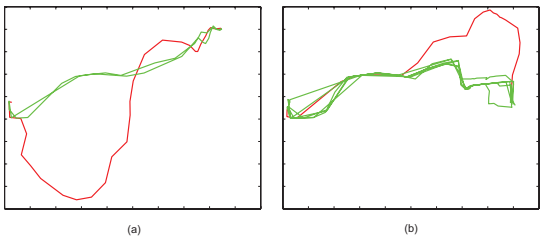


Figure 7: The Mechanism for Handling Excuses.

To further understand the regularity of driving fraud activities, we explore the temporal distribution of all driving fraud activities in Figure 6, where the temporal unit is an hour. In the figure, each dot point represents a driving fraud activity with corresponding driver on x-axis and hour on y-axis. We highlight top-3 drivers with the most fraud activities as vertical blue lines and top-4 time zones (hours) containing most fraud activities as horizontal blue lines. Here, we can observe that more fraud activities are committed around 5PM, 7PM, 8PM and 10PM.

3.2 Mechanism for Possible Excuses

To deal with the possible excuses, we introduce one mechanism to our system. Specifically, after we detect a suspicious driving activity, we recall all previous driving traces of the corresponding driver to check if the driver is familiar with the roads. Meanwhile, to confirm the traffic-related excuse, we recall all driving traces around the same timestamp. For example, in Figure 7 (a), after detecting the driving fraud activity (red one), we recall and plot the drivers' previous driving trajectories (green ones) and can clearly find this driver often operates in this area. Also, in Figure 7 (b), we show all trajectories (green ones) happening within the same minute as the driving fraud activity (red one) and we can see many other drivers did not detour. By this mechanism, we are able to obtain more real-time evidences to deny possible excuses and confirm the fraud behaviors.

4. BUSINESS INSIGHT EXPLORATION

In this section, we briefly introduce the third function, which is designed to exploit taxi data to discover some business insights, which may benefit taxi drivers or companies.

First, we examine the tip geo-distribution. With the taxi data, we are able to obtain the fare and tip information for each transaction, i.e., passenger delivery from a pick-up location to a drop-off location. Considering the different amount of fare for individual transaction, we use relatively tip, which is equal to the ratio of tip to fare, instead of the original tip value. After grouping the transactions, which are spatially neighboring, we obtain the average relative tip in the group and the rough tip geo-distribution across the city. Some examples of geo-distribution are accessible in the online demonstration system.

Also, we introduce the concept of economic potential to evaluate each pick-up point (cluster). The economic potential is expected to take into account not only the probability of pick-up events, but also the expected earnings of transactions (delivery) starting from the pick-up point. Furthermore, the economic potential also contains the "sustainability" information. For example, some pick-up points may usually lead to some very far destinations, where few customers are there. Experienced drivers may be not inclined to take this kind of delivery. Due to the space limitation, we are going to illustrate this economic potential concept in our online demonstration system.

5. CONCLUSION AND FUTURE WORK

In summary, we demonstrated a taxi business intelligence system, which is capable to exploit the knowledge from taxi data for business use. Indeed, the discovered knowledge can be helpful in many different applications, such as improving taxi business performances and passenger experiences. Specifically, in this system, we implemented three functions: route recommendation, fraud detection, and taxi business insight exploration. We developed the system with two sets of real-world taxi data, which were collected from taxis in San Francisco and the New York City. Some preliminary results of three functions are shown in this paper. More results can be explored using the online demo system.

In the future, we plan to improve route recommendation from both efficiency and effectiveness perspectives. Also, more business insights are expected to be mined. In addition, we are communicating with some taxi agents for both more taxi data and exploiting the system in their business.

6. ACKNOWLEDGEMENTS

This research was partially supported by National Natural Science Foundation of China (NSFC) via project numbers 70890082 and 71028002. Also, this work was partly sponsored by WINLAB with industry affiliation program with Panasonic in pursuing data mining to abnormal behavior detection for security and safety applications.

7. REFERENCES

[1] http://www.bustathief.com/taxi-fraud-taxi-scam/.
[2] http://www.consumertraveler.com/today/nyc-taxi-drivers-overcharge-passengers-8-3-millioin/.
[3] http://www.tour-beijing.com/taxi/.
[4] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. J. Pazzani. An energy-efficient mobile recommender system. In *SIGKDD*, 2010.
[5] Y. Ge, H. Xiong, Z.-H. Zhou, H. Ozdemir, J. Yu, and K. Lee. Top-eye: Top-k evolving trajectory outlier detection. In *ACM CIKM*, Toronto, 2010.
[6] D. Grosu and A. T. Chronopoulos. Algorithmic mechanism design for load balancing in distributed systems. *IEEE TSMC-B*, 34(1):77–84, 2004.
[7] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *SIGKDD*, pages 444–452, Nevada, USA, 2008.
[8] Z. Xu and R. Huang. Performance study of load balancing algorithms in distributed web server systems. In *TR*, CS213 Univ. of California, Riverside.