2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

Recommend a Profitable Cruising Route for Taxi Drivers

Hao Dong, Xuedan Zhang, Yuhan Dong*, Chuang Chen, Fan Rao
Department of Electronic Engineering, Tsinghua University, Beijing, China
Email: photodh@126.com, xuedanzhang@gmail.com, yhdong@gmail.com

***Abstract*—Every time after the taxi drivers drop off their previous passengers, they have to decide how to search for the next passengers. It may lead to poor performance if the taxi drivers cruise according to their own experience, especially for the freshmen. Therefore, it is valuable to recommend a profitable cruising route for the taxi drivers in order to increase their income and reduce waste in fuel. In this paper, we propose to use a system of linear equations to calculate the score of each road segment based on a large-scale real-world GPS data set. The score of each road segment consists of 1) the total income of the road segment and 2) the attractiveness of the drop-off location with respect to the next pick-up. Then, we get the profitable cruising route based on the score of each road segment using skyline computation. We build our system using historical data set generated by 12,000 taxis of Beijing in November 2012. Finally, we demonstrate that taxi driver who follows our recommendation can enhance their income compared to the ground truth.**

I. Introduction

In many big cities, like New York, Beijing and Singapore, taxis are equipped with GPS sensors for dispatching, navigation and localization. Typically, one taxi will report on its present location, timestamp and status (occupied, cruising, etc) to data center in a certain frequency. Through these large-scale real-world GPS data, we could conduct lots of researches to better understanding our city, such as predictability of human mobility [17], urban computing [10] [13], route planning [20], anomaly detection [23], carpooling [25], passenger/taxi-finding strategies [22] [26].

Taxi service has a quite significant role in public transportation service nowadays. According to a recent survey [19], there are 41% of the people would take a taxi weekly, and 25% of the people would take a taxi daily in New York. It is necessary to find an efficient and profitable passenger-finding strategy for taxi drivers. An experienced taxi driver knows how to plan his/her own routes after dropping off a passenger, but not an inexperienced one. On the other hand, the experienced taxi driver may not be familiar with one entire big city, like Beijing. Fig. 1 shows the percentage of occupied miles at each hour of one day. From this figure, we can see that the percentage of occupied miles is only 60%-70% even in rush hours. Therefore, we would use a profitable cruising route to help taxi drivers find their next passengers in this paper. We not only want the taxi to pick up a passenger as quickly as possible, but also expect the passenger that the taxi picked up to be more profitably.

*Dr.Yuhan Dong is the corresponding author.

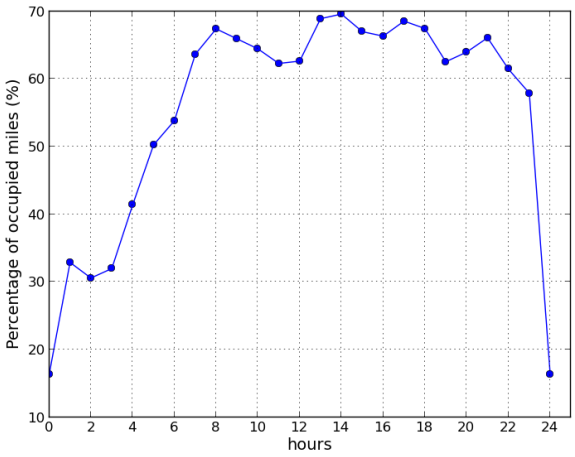

Fig. 1. Percentage of occupied miles

The main contribution of this paper is that we use a system of linear equations to calculate the score of each road segment. The score of each road segment is influenced by the following three factors:

- 1) The road segment where he/she can pick up passenger with largest probability.
- 2) The length of the occupied paths that originate from the road segment.
- 3) The occupied paths that originate from the road segment have many drop-off points scattered in different road segments. The attractiveness of these drop-off location with respect to the next pick-up would be considered.

After we get the score of each road segment, we would recommend a profitable cruising route which has maximal score to the taxi drivers. Considering the complexity and accuracy, skyline computation is introduced to solve this problem.

The rest of the paper is organized as follows. In the next section, we review the related works of passenger-finding strategies. In section III, we introduce the taxi GPS trace data set and state some basic definitions. In section IV, we present the framework of our passenger-finding strategy. In section V, a system of linear equations is proposed to calculate the score of each road segment more credibly. In section VI, we use skyline computation to recommend a profitable cruising route for taxi driver based on the score of each road segment.

978-1-4799-6078-1/14/$31.00 ©2014 IEEE

In section VII, we validate our work. Section VIII concludes this paper.

II. Related Works

Passenger-finding strategies have been investigated by a number of groups. Most of the papers are focused on recommending hotspot to the cruising taxi driver. Chang et al [3] extract hotspots from large amounts of pick up points using clustering algorithm. Then, it would recommend a hotspot to taxi driver based on the hotness of the hotspot. Similarly, [8] propose an improved ARIMA method to forecast the spatial-temporal variation of passengers in a hotspot to help taxi drivers find new passengers. [7] states that a taxi driver would remain in a local area (waiting) or travel a longer distance (hunting) to find a new passenger. Then, L1-Norm Support Vector Machine (SVM) is used to determine whether the taxi driver should wait or hunt based on the current location and time.

Instead of recommending one hotspot around the taxi driver, [5] [6] [27] are focused on how to recommend a sequence of hotspots to maximize the probability of picking up a passenger with less cruising distance. In [22], the authors propose an algorithm to extract parking places based on the distance between non-occupied trajectories' points. Then, it aims to provide the taxi driver with the best parking place and the best route to this parking place.

In [15] [18], the authors first find pick-up points from large-scale real-world GPS data set using clustering algorithm and calculate the success probability of each pick-up point. Then, it assumes that the taxi driver will wait at the current pick-up point or move to the next pick-up point. In this occasion, an Markov Decision Process (MDP) model is proposed to solve this problem.

However, there are few works concentrate on how to cruise for the next passengers instead of recommending hotspots to the taxi drivers. It is worth to mention that cruising route recommendation is more specifically and accurately. In [14], a Spatio-Temporal Profitability (STP) map is constructed based on historical data to guide the taxi drivers in terms of dividing a region into a grid of equally sized cells. In [24], a system called *pCruise* is proposed to reduce the taxi's cruising mile. *pCruise* uses cruising graph to model a taxi's cruising process, where vertices represent intersections and edges represent road segments connecting intersections. Then, *pCruise* would recommend a shortest cruising route with at least one expected available passenger for the taxi driver based on cruising graph by an efficient sheduling. [4] formalizes the passenger-finding strategies into a new problem, global-optimal trajectory retrieving. In order to solve this problem, a system called *HUNTS* is presented.

The closest work to ours is [4]. In [4], *HUNTS* uses a dynamic scoring system to calculate the score of each road segment which takes picking-up rate and average income into consideration. We would like to point out that picking-up rate can be inaccurately when the historical data is sparse in one road segment. For example, if there are only one taxi passing by the road segment and this taxi pick up a passenger. Then, the picking-up rate would be 100% which wouldn't be realistic to represent the actual picking-up rate. On the other hand, our recommendation considered the attractiveness of the drop-off location with respect to the next pick-up. Therefore, our work can obtain more rewarding in the long term.

III. Data Description and Basic Definitions

A. Data Description

We get the taxi GPS trace data set which are used in this paper from datatang [1]. This data set includes more than 1 billion records which were collected by 12,000 taxis of Beijing in November 2012. There are about 4,300,000 occupied paths and 5,000,000 cruising paths. Most of the taxis' sampling rate is approximately 1 minute and an extra sampling is performed when the status changes, i.e. a passenger is picked up or dropped off. Each records consists of the following fields.

- Taxi ID: unique id of the taxi
- Time: the sampling time, in the format of 'YYYY-MM-DD HH:MM:SS'
- Longitude: the longitude of the taxi
- Latitude: the latitude of the taxi
- Status: the status of the taxi, including occupied (represent as 1) and cruising (represent as 0).

B. Basic Definitions

Definition 1 (Road Segment): A road segment is a directed edge r that is associated with an id $r.wid$, a starting point $r.s$, an ending point $r.e$ and a list of intermediate points. Another important parameter is the length of road segment and represent as $r.l$. It is a remarkable fact that a street is consist of multiple road segments.

Definition 2 (Road Network): The road network is a digraph $G(V, E)$, where V is a set of vertices representing the endpoints of a road segment. All the road segments are included in E, and there are $||E||$ road segments.

Definition 3 (Cruising Route): A cruising route is a set of road segments $\varsigma = \{r_1, ..., r_i, r_j, ..., r_n\}$, where $r_i.e = r_j.s$. The length of a cruising route $\varsigma.l$ is the sum of these road segments' length, i.e. $\varsigma.l = \sum_{k=1}^{n} r_k.l$

Definition 4 (Status): The status indicates whether a taxi has a passenger or not. The taxi is occupied if it has a passenger and represent as 1. The taxi is cruising if it is traveling without a passenger and represent as 0.

Definition 5 (Trajectory): A trajectory τ is a sequence of GPS points with same status except the last one. The starting point of a trajectory is $\tau.s$, and the ending point of a trajectory is $\tau.e$. The distance between a point p to a road segment r is defined as the minimum Euclid distance from p to the point on r, e.g., $d(p, r) = min_{q \in r} d(p, q)$.

Definition 6 (Path): After map matching, a trajectory τ would convert to a path ρ which is denoted by a sequence of road segments r_i, where $i = 1, 2, ..., ||\rho||$. $||\rho||$ represents

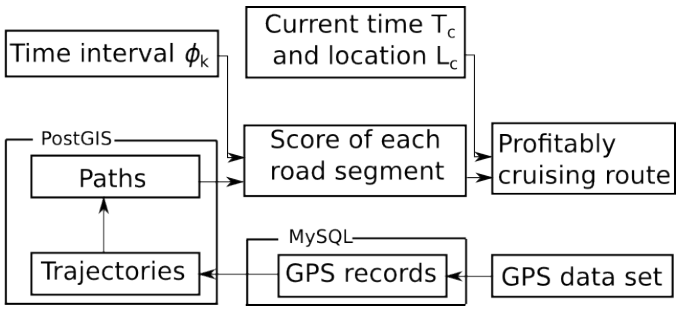

Fig. 2. The framework of our system

the number of road segments that ρ traverses through. The starting point of a path ρ is $\rho.s$ which is the projection point of $\tau.s$ on r_1. The ending point of a path ρ is $\rho.e$ which is the projection point of $\tau.e$ on $r_{||\rho||}$.

Definition 7 (Path Length): The length of a path is denoted by $\rho.l$. $\rho.l$ is the length between $\rho.s$ and $\rho.e$ on the given path ρ.

IV. System Overview

Fig. 2 shows the framework of our system. First, we would put all the GPS records that are included in our GPS data set into MySQl. Then, we could extract the occupied trajectories from these GPS records and dump into PostgreSQL as PostGIS objects. Until now, ther are several map-matching algorithms have been proposed, such as [11] [12] [21] [9]. In this paper, we use the ST-Matching [11] to map-match these occupied trajectories to paths, and these paths are dumped back into PostgresSQL.

We partition one day into several time intervals, and the length of each time period is ψ. Then, we could calculate the score of each road segment in each time period. In this paper, we set ψ to two hours. The kth time interval is:

$$\phi_k = [(k-1)\psi, k\psi], k = 1, 2, ..., K \tag{1}$$

We could select all the occupied paths which belong to time interval ϕ_k. Then, we would use a system of linear equations to calculate the score of each road segment based on the previously selected paths. In other time intervals, we could use the same process to calculate the score of each road segment. Given the current time T_c and location L_c of a taxi driver, we would recommend a profitably cruising route to he/she based on the score of each road segment. The profitably cruising route should have maximum score within a certain length.

V. Calculate the Score of Each Road Segment

The score of one road segment can indicates how much this road segment is suitable for taxi driver to cruise for a passenger. In this section, we demonstrate how to calculate the score of each road segment.

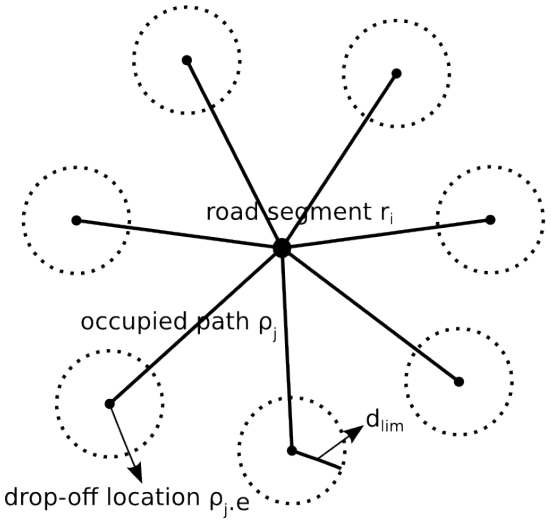

Fig. 3. The score of road segments r_i

Given an occupied path ρ and its length $\rho.l$, we would use the following Equation (2) to calculate the income $f(\rho)$.

$$f(\rho) = \begin{cases} 10 & \rho.l \leq 3km \\ 10 + 2 * (\rho.l - 3) & 3km < \rho.l \leq 15km \\ 34 + 3 * (\rho.l - 15) & 15km < \rho.l \end{cases} \tag{2}$$

In Fig. 3, we illustrate how to calculate the score of road segment r_i. From Fig. 3, we can see that there are several occupied paths which start from r_i. For each occupied path ρ_j, we can get the income of ρ_j by $f(\rho_j)$ and the attractiveness of drop off location $\rho_j.e$ with respect to the next pick-up. We would use Equation (3) to calculate the attractiveness $a(\rho_j.e)$ of drop off location $\rho_j.e$ with respect to the next pick-up. In Equation (3), x_k is the score of road segment r_k. The drop off location of each occupied path ρ_j would have a significant impact on whether the taxi driver could pick up a profitable passenger quickly in the later deal or not. Therefore, the drop off locations are better, the score of r_i is higher.

$$a(\rho_j.e) = \sum_{r_k \in E} w(d(\rho_j.e, r_k))x_k \tag{3}$$

In Equation (3), $w(d(\rho_j.e, r_k))$ is the weight of x_k corresponding to the distance $d(\rho_j.e, r_k)$ between $\rho_j.e$ and r_k. We use the following Equation (4) to get $w(d(\rho_j.e, r_k))$. In this paper, d_{lim} is set to $3km$.

$$w(d(\rho_j.e, r_k)) = \begin{cases} 1 & d(\rho_j.e, r_k) \leq d_{lim} \\ 0 & d(\rho_j.e, r_k) > d_{lim} \end{cases} \tag{4}$$

In order to simplify the calculation, we use an approximation $\hat{d}(\rho_j.e, r_k)$ to substitute $d(\rho_j.e, r_k)$. As show in Fig. 4, $\rho_j.e$ belongs to road segment r_p. We would like to say that $\hat{d}(\rho_j.e, r_k)$ is the minimum value among d_1, d_2, d_3 and d_4,

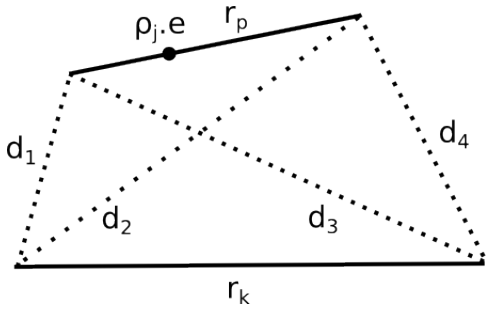

Fig. 4. Distance between $\rho_j.e$ and r_k

which is illustrate in Fig. 4. In this way, we could calculate $d(\rho_j.e, r_k)$ throuhg Equation (5).

$$d(\rho_j.e, r_k) = \hat{d}(\rho_j.e, r_k) = min(d_1, d_2, d_3, d_4) \quad (5)$$

We use $P = \{\rho_i\}$ to represent the set of occupied paths that belongs to a certain time interval ϕ_k. For a road segment r_i, we denote the score by x_i. The score of each road segment at a certain time interval can be calculated by Equation (6).

$$\begin{aligned} x_i &= \sum_{\substack{\rho_j \in P \\ \rho_j.s \in r_i}} [\alpha f(\rho_j) + \beta a(\rho_j.e)] \\ &= \sum_{\substack{\rho_j \in P \\ \rho_j.s \in r_i}} [\alpha f(\rho_j) + \beta \sum_{r_k \in E} w(d(\rho_j.e, r_k)) x_k] \end{aligned} \quad (6)$$

The first part of Equation (6) represent the total income of road segment r_i. The second part of Equation (6) represent the attractiveness of drop-off location with respect to the next pick-up. The parameters α and β are coefficient of each part. In this paper, we set α and β to 0.5.

Let $P_i = \{\rho_j | \rho_j \in P, \rho_j.s \in r_i\}$ be the set of occupied paths which start from r_i. Then, we can simplify Equation (6) to Equation (7) where $w_{\rho_j.e,k} = w(d(\rho_j.e, r_k))$.

$$x_i = \alpha \sum_{\rho_j \in P_i} f(\rho_j) + \beta \sum_{\rho_j \in P_i} \sum_{r_k \in E} w_{\rho_j.e,k} * x_k \quad (7)$$

If Equation (8) is satisfied, x_i would be convergent according to the theorem of linear equations.

$$\beta \sum_{\rho_j \in P_i} \sum_{r_k \in E} w_{\rho_j.e,k} < 1 \quad (8)$$

Let $\mathbf{X} = [x_1, ..., x_i, ..., x_{||E||}]$ be a vector of the score of every road segment. Then we could get a system of linear equations from Equation (7), which can be solved by generalized minimal residual method (GMRES) [16].

VI. Recommend a Profitable Cruising Route Using Skyline computation

After the taxi driver drop off a passenger on location L_c at time T_c, he/she would need a route to cruise in order to pick up next passenger. In this section, we demonstrate how to get a profitable cruising route for taxi driver based on the score of each road segment in a certain time interval.

We could get the score of each road segment in the time interval ϕ_k using the method we proposed in section IV, where $T_c \in \phi_k$. In this case, a profitable cruising route is equivalent to a max-score cruising route. In other words, we would recommend a cruising route ς_{max} starting from r_c ($L_c \in r_c$) to taxi driver which has maximal score while $\varsigma_{max}.l \leq l_{lim}$. We use l_{lim} to denote the length limit of a recommended cruising route and we set it to $3km$ in this paper.

In the following, we give the formal definition of max-score cruising route. We use $\varsigma.x$ to denote the score of a cruising route ς.

Definition 8 (Max-Score Cruising Route (MSCR)): Given the current location L_c and time T_c of a taxi driver where $L_c \in r_c$ and $T_c \in \phi_k$, we would recommend a cruising route $\varsigma_{max} = \{r_c, ..., r_i, ..., r_n\}$ for taxi driver which has maximum score $\varsigma_{max}.x = \sum_{i=c}^{n} x_i$ under the limit of $\varsigma_{max}.l \leq l_{lim}$.

The max-score cruising route problem is NP-complete. We proposed an heuristic algorithm which is show in Algorithm 1 to solve this problem. In Algorithm 1, we use skyline computation to reduce the search space. The method of skyline computation is adopted by [4] [5] [6].

Algorithm 1 Max-Score Cruising Route (MSCR)

```
Input: The score vector of all the road segments X = {x_1, ..., x_i, ...x_||E||}, Length limit l_lim, Current road segment r_c
Output: The max-score cruising route ς_max
ς_max ← [r_c]
ςs ← [ς_max], ς̂s ← [ ]
while ςs ≠ ς̂s do
    ς̂s ← ςs
    for each route ς ∈ ςs do
        for each subsequent road segment ς.sub of ς do
            ς̂ ← ς ⋃ r_sub
            if ς̂ ≤ l_lim then
                ς̂s ← ς̂s ⋃ ς̂
                delete ς from ς̂s
            end if
        end for
    end for
    if ς_i dominate ς_j, where ς_i, ς_j ∈ ςs then
        remove ς_j from ςs
    end if
end while
find the max-score cruising route ς_max ∈ ςs
return ς_max
```

Definition 9 (Cruising Route Dominance): Let $\varsigma s = \{\varsigma_1, ..., \varsigma_i, ..., \varsigma_j, ..., \varsigma_n\}$ denote a set of cruising routes. Each of these cruising route can be described by two dimensions

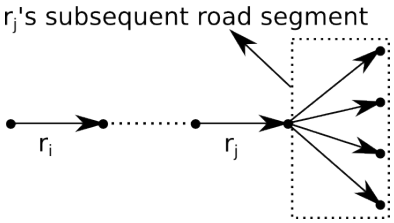

Fig. 5. From ς to $\hat{\varsigma}$, $\varsigma = \{r_c, ..., r_j\}$

$\varsigma_i = (\varsigma_i.l, \varsigma_i.x)$. We define that ς_i dominates ς_j iff the following cases happens: (1) $\varsigma_i.x > \varsigma_j.x$ and $\varsigma_i.l \leq \varsigma_j.l$; (2) $\varsigma_i.x = \varsigma_j.x$ and $\varsigma_i.l < \varsigma_j.l$.

The pseudocode of calculating the max-score cruising route problem is given in Algorithm 1. Let $\mathbf{X}$ be the score vector of every road segment. We initialize ς_{max} to r_c and ςs to ς_{max}. ςs is a set of cruising route and $\hat{\varsigma s}$ is a buffer used in Algorithm 1.

This algorithm is iterative. In each iteration, we get a copy of ς_s and put it in $\hat{\varsigma s}$. For each route ς in ςs, if the length of ς plus the length of one subsequent road segment of ς is less than l_{lim}, then we would put the new route $\hat{\varsigma}$ into $\hat{\varsigma s}$. Once a route $\hat{\varsigma}$ is added to $\hat{\varsigma s}$, we should delete the route ς from $\hat{\varsigma s}$.

Fig. 5 illustrate the procedure of constructing $\hat{\varsigma}$ from ς. We assume that $\varsigma = \{r_c, ..., r_j\}$. Then, the last road segment of route ς is r_j. From the digraph $G(V, E)$, we can find the subsequent road segments $r_j.sub$ of r_j. We would like to proclaim that the subsequent road segments $\varsigma.sub$ of ς is equivalent to $r_j.sub$.

Some routes from ςs are discarded in line 16 using skyline computation. It discards some routes in ςs which are dominated by another route in ςs. According to **Definition** 9, we can find out whether ς_j is dominated by ς_i, where $\varsigma_i, \varsigma_j \in \varsigma s$. Such iteration is repeated until no extra route is added to $\hat{\varsigma s}$, i.e., $\varsigma s = \hat{\varsigma s}$. Then, we can get the score of all the cruising routes in ςs and find the max-score cruising route $\varsigma_{max} \in \varsigma s$.

VII. Validation

A. Experiment Setup

In this paper, we regard the data from Nov. 1st 2012 to Nov. 20th 2012 as the training, and use the data from Nov. 20th 2012 to Nov. 30th 2012 for testing. There are 400 thousands occupied paths between 8:00 and 10:00 from the training data set. We would use these occupied paths to calculate the score of each road segment. There are a large amount of road segments which pick up less than one passenger in one day between 8:00 and 10:00, and we set these road segments' score to zero. The digital map of Beijing has 49,122 road segments, and 5,212 road segments' score would not be zero.

B. Performance Comparison

Fig. 6 shows the importance of drop-off locations. There are 20 occupied paths start from road segment r_1 ($r_1.wid =$ 30762) and r_2 ($r_2.wid =$ 21493). Almost all the occupied paths which start from r_1 head towards suburb. Most of the occupied paths which start from r_2 head towards downtown. We can tell that the drop-off locations of occupied paths which start from r_2 is better than r_1's. As well as that the total income of r_1 is similarly to r_2. Therefore, the road segment r_2 is better than road segment r_1. In conclusion , it is necessary to consider the importance of drop off locations which lead to that r_2's score is much higher than r_1's.

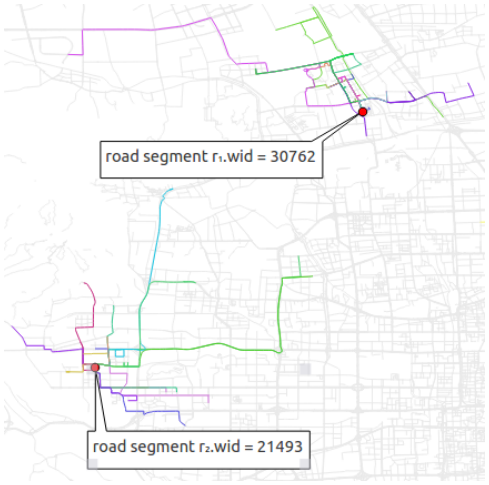

Fig. 6. The importance of drop-off locations

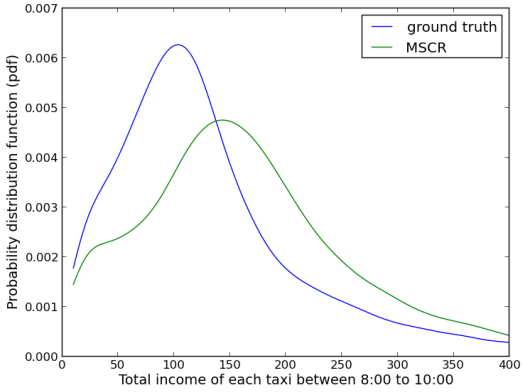

Fig. 7. The distribution of taxi drivers' income

We can get the occupied paths between 8:00 to 10:00 from the testing data set of each taxi. Then, the income of each taxi can be calculated and the probability distribution function (pdf) of ground truth is show in Fig. 7. We compare the proposed MSCR with ground truth to validate that our system indeed improve the income of taxi drivers. We first find the cruising location L_c and time T_c of each taxi which starts cruise near 8:00 each day. Then, We could recommend a cruising route ς_{max} to this taxi. We assume that the taxi would pick up a passenger in the road segment r_{max} which has max score in ς_{max}. There are several occupied paths which start from r_{max} and we would randomly select one to

give the taxi driver. It is worth mentioning that we would add 5 minutes in each deal for punishment. We use the drop off location and the time interval of the selected occupied path plus T_c as the next cruising location and time, respectively. This procedure would continue until the time is later than 10:00. We could get the income of each taxi which follows our recommended cruising route and also plot the pdf in Fig. 7. From Fig. 7 we could see that our system could make the taxi drivers earn more money compared with ground truth.

VIII. Conclusion

With the availability of large-scale real-world GPS data set, we can better understand our city and dig up some useful information. There is a survey [2] which summarize the previous research work using the massive GPS data set. In this paper, we construct a system for the taxi driver to find a profitable passenger using historical data set generated by 12,000 taxis of Beijing in November 2012. We use a system of linear equations to calculate the score of each road segment.We first consider the importance of drop-off locations of every occupied paths which start from one road segment.

Many taxi drivers are begin to use the taxi booking apps (such as Kuaidi Dache or Didi Dache) to find passengers nowadays. With the use of taxi booking apps, the locations of currently waiting passengers is known in advance. When there are several waiting passengers called through the taxi booking apps, we can recommend a taxi driver to pick up a passenger based on the distance to that passenger and the drop off location of the deal. However, it is more practically to recommend a waiting point for the taxi drivers when there are no passenger is called through the taxi booking apps. The taxi booking apps become more and more popular which makes the passenger-finding strategies changed dramatically. In the future, we will study the passengers-finding strategies combine large-scale real-world GPS data set and taxi booking apps.

References

[1] *http://www.datatang.com/data/44502*.

[2] Pablo Samuel Castro, Daqing Zhang, Chao Chen, Shijian Li, and Gang Pan. From taxi gps traces to social and community dynamics: a survey. *ACM Computing Surveys (CSUR)*, 46(2):17, 2013.

[3] Han-wen Chang, Yu-chin Tai, and Jane Yung-jen Hsu. Context-aware taxi demand hotspots prediction. *International Journal of Business Intelligence and Data Mining*, 5(1):3–18, 2010.

[4] Ye Ding, Siyuan Liu, Jiansu Pu, and Lionel M Ni. Hunts: A trajectory recommendation system for effective and efficient hunting of taxi passengers. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 1, pages 107–116. IEEE, 2013.

[5] Yong Ge, Hui Xiong, Alexander Tuzhilin, Keli Xiao, Marco Gruteser, and Michael Pazzani. An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 899–908. ACM, 2010.

[6] Haoran Hu, Zhiang Wu, Bo Mao, Yi Zhuang, Jie Cao, and Jingui Pan. Pick-up tree based route recommendation from taxi trajectories. In *Web-Age Information Management*, pages 471–483. Springer, 2012.

[7] Bin Li, Daqing Zhang, Lin Sun, Chao Chen, Shijian Li, Guande Qi, and Qiang Yang. Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 63–68. IEEE, 2011.

[8] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, 6(1):111–121, 2012.

[9] Yang Li, Qixing Huang, Michael Kerber, Lin Zhang, and Leonidas Guibas. Large-scale joint map matching of gps traces. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 214–223. ACM, 2013.

[10] Yu Liu, Chaogui Kang, Song Gao, Yu Xiao, and Yuan Tian. Understanding intra-urban trip patterns from taxi trajectory data. *Journal of geographical systems*, 14(4):463–483, 2012.

[11] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 352–361. ACM, 2009.

[12] Takayuki Osogami and Rudy Raymond. Map matching with inverse reinforcement learning. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2547–2553. AAAI Press, 2013.

[13] Gang Pan, Guande Qi, Zhaohui Wu, Daqing Zhang, and Shijian Li. Land-use classification using taxi gps traces. *Intelligent Transportation Systems, IEEE Transactions on*, 14(1):113–123, 2013.

[14] Jason W Powell, Yan Huang, Favyen Bastani, and Minhe Ji. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *Advances in Spatial and Temporal Databases*, pages 242–260. Springer, 2011.

[15] Shiyou Qian, Yanmin Zhu, and Minglu Li. Smart recommendation by mining large-scale gps traces. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 3267–3272. IEEE, 2012.

[16] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[17] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[18] Haochen Tang, Michael Kerber, Qixing Huang, and Leonidas Guibas. Locating lucrative passengers for taxicab drivers. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 494–497. ACM, 2013.

[19] N.Y.C. Taxi and L. Commission. Taxi of tomorrow survey results. 2011.

[20] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. T-drive: Enhancing driving directions with taxi drivers' intelligence. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):220–232, 2013.

[21] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. An interactive-voting based map matching algorithm. In *Mobile Data Management (MDM), 2010 Eleventh International Conference on*, pages 43–52. IEEE, 2010.

[22] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, and Xing Xie. T-finder: A recommender system for finding passengers and vacant taxis. *Knowledge and Data Engineering, IEEE Transactions on*, 25(10):2390–2403, 2013.

[23] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. ibat: detecting anomalous taxi trajectories from gps traces. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 99–108. ACM, 2011.

[24] Desheng Zhang and Tian He. pcruise: Reducing cruising miles for taxicab networks. In *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*, pages 85–94. IEEE, 2012.

[25] Desheng Zhang, Tian He, Yunhuai Liu, and John A Stankovic. Callcab: A unified recommendation system for carpooling and regular taxicab services. In *Big Data, 2013 IEEE International Conference on*, pages 439–447. IEEE, 2013.

[26] Xudong Zheng, Xiao Liang, and Ke Xu. Where to wait for a taxi? In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 149–156. ACM, 2012.

[27] Qingnan Zou, Guangtao Xue, Yuan Luo, Jiadi Yu, and Hongzi Zhu. A novel taxi dispatch system for smart city. In *Distributed, Ambient, and Pervasive Interactions*, pages 326–335. Springer, 2013.