Where to Find My Next Passenger?

Jing Yuan[1,2], **Yu Zheng**[2], **Liuhang Zhang**[1,2], **Xing Xie**[2] **and Guangzhong Sun**[1]
[1]University of Science and Technology of China
[2]Microsoft Research Asia
yuanjing@mail.ustc.edu.cn, {yuzheng,xing.xie}@microsoft.com

ABSTRACT
We present a recommender for taxi drivers and people expecting to take a taxi, using the knowledge of 1) passengers' mobility patterns and 2) taxi drivers' pick-up behaviors learned from the GPS trajectories of taxicabs. First, this recommender provides taxi drivers with some locations (and the routes to these locations), towards which they are more likely to pick up passengers quickly (during the routes or at the parking places) and maximize the profit. Second, it recommends people with some locations (within a walking distance) where they can easily find vacant taxis. In our method, we propose a parking place detection algorithm and learn the above knowledge (represented by probabilities) from trajectories. Then, we feed the knowledge into a probabilistic model which estimates the profit of a parking place for a particular driver based on where and when the driver requests for the recommendation. We validate our recommender using trajectories generated by 12,000 taxis in 110 days.

Author Keywords
Taxicab, recommender, parking place

ACM Classification Keywords
H.2.8 Database Management: data mining, spatial databases and GIS.

General Terms
Algorithms, Design, Experimentation, Performance

INTRODUCTION
Taxicabs play an important role in people's commute between public and private transports. A significant number of people are traveling by taxis in their daily lives around the world. According to a recent survey about the taxi service of New York City [7], 41% people take a taxi per week and 25% of the respondents take a taxi everyday. However, on one hand, to facilitate people's travel, major cities, like New York, Tokyo, London, and Beijing, have a huge number of taxis traversing in urban areas. The vacant taxis cruising on roads not only waste gas and time of a taxi driver but also generate additional traffic in a city. Thus, how to improve the utilization of these taxis and reduce the energy consumption effectively poses an urgent challenge. On the other hand, many people feel frustrated and anxious when they are unable to find a taxicab after waiting for a long time.

To address this issue, we propose a recommender for both taxi drivers and passengers using a huge number of historical GPS trajectories of taxis. Specifically, on one hand, given the geo-position and time of a taxicab looking for passengers, we suggest the taxi driver with a location, towards which he/she is most likely to pick up a passenger as soon as possible and maximize the profit of the next trip, as demonstrated in Figure 1 A). This recommendation helps to reduce the cruising (without a fare) time of a taxi thus saves energy consumption and eases the exhaust pollution as well as helps the drivers to make more profit. On the other hand, we provide people expecting to take a taxi with the locations (within a walking distance) where they are most likely to find a vacant taxicab, as shown in Figure 1 B). Using our recommender, a taxi will find passengers more quickly and people will take a taxi more easily; therefore, reduces the above-mentioned problem to some extent.

A) Taxi recommender B) Passenger recommender

Figure 1. Recommendation scenario

Recently, in many big cities, like New York, Beijing and Singapore, taxicabs are equipped with GPS sensors for dispatching and safety. Typically, these taxis will report their present locations to a data center in a certain frequency, e.g., 2 minutes [12]. Besides a geo-position and timestamp, the occupancy information of a taxi is also recorded (using some weight sensor or by connecting a taxi meter with the embedded GPS device). Therefore a large number of such GPS trajectories with occupancy information are being generated everyday. Intuitively, these taxi trajectories contain two aspects of knowledge. One is passengers' mobility, i.e., where and when passengers get on and off a taxi. The other is taxis' pick-up behaviors. For example, where the high-profit taxi drivers usually go and how they can find passengers quickly. With these two aspects of knowledge, we can recommend

UbiComp'11, September 17–21, 2011, Beijing, China.

locations with high-probability to pick up a passenger to the taxi driver and suggest locations where a passenger is easy to find a vacant taxi.

The major contribution of this work lies in the following aspects:

- We propose an approach for accurately detecting *parking places* from the GPS trajectories of a large number of taxis. These parking places stand for the locations where taxi drivers usually wait for passengers with their taxi parked. From these parking places, we can calculate the probability of picking up a passenger if the driver goes towards a parking place (including the situation that the driver picks up a passenger when cruising), hence, enable the recommender for taxi drivers.
- We devise a probabilistic model to formulate the time-dependent taxi behaviors (pick-up/drop-off/cruising/parking), both on road segments and in parking places, based on which, we build the recommendation solution for taxi drivers and passengers. We devise a partition-and-group framework to learn the citywide statistical knowledge so as to provide just-in-time recommendations with time varying information learned from the historical data.
- We evaluate our method using a large number (12,000 taxis during 110 days) of historical GPS trajectories generated by taxicabs. The evaluation results validate that our method can effectively suggest the taxi drivers with locations towards which the driver can make more profit and save cruising time.

OVERVIEW

Preliminary

DEFINITION 1 (ROAD SEGMENT). A *road segment* r is a directed edge that is associated with a direction symbol $r.dir$ (one-way or bidirectional), two terminal points $r.s$ and $r.e$, road level $r.level$, as well as the travel time $r.t$.

DEFINITION 2 (ROUTE). A *route* R is a sequence of connected road segments, i.e., R: $r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_n$, where $r_{k+1}.s = r_k.e$, $(1 \leq k < n)$. The start point and end point of a route can be represented as $R.s = r_1.s$ and $R.e = r_n.e$.

DEFINITION 3 (STATE). We consider three states for a working taxi: *occupied* ($\mathcal{O}$), *cruising* ($\mathcal{C}$) and *parking* ($\mathcal{P}$), detailed in Table 1. The taxi is *non-occupied* for both cruising and parking states.

State	Taxi Status
Occupied ($\mathcal{O}$)	A taxi is occupied by a passenger.
Cruising ($\mathcal{C}$)	A taxi is traveling without a passenger.
Parking ($\mathcal{P}$)	A taxi is waiting for a passenger.

Table 1. The states of a taxi

Note that the "parking" state proposed in this paper is the status that taxi drivers wait somewhere for business, i.e., stay and/or queue for a while with the intention to get a passenger on-board. This status is frequently found at airports, hotels, shopping centers, etc. We call these places where the taxis frequently wait for passengers as *parking places*. Note the parking place here does not merely imply a parking lot for private vehicles (which is the typical definition for "parking place").

DEFINITION 4 (TRAJECTORY and TRIP). A taxi *trajectory* is a sequence of GPS points logged for a working taxi, where each point p has the following fields: time stamp $p.t$, latitude $p.lat$, longitude $p.lon$, located road segment (provided by map matching algorithms [13]) $p.r$, state $p.s$ (The raw GPS trajectory only indicates whether a point is occupied or non-occupied). A taxi *trip* is a sub-trajectory which has a single state, either cruising (need to be inferred) or occupied. Refer to Figure 2 for an example. Note that a taxi could generate multiple trips between two parking places.

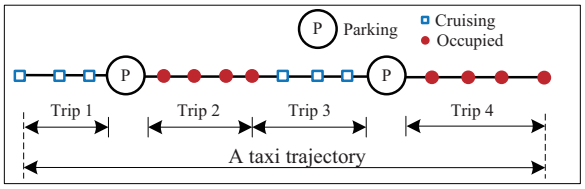
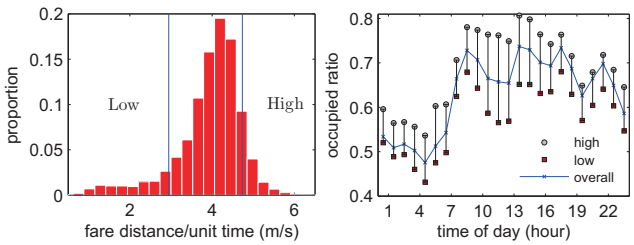


Figure 2. Taxi trajectory and taxi trip

Motivation

Different from other public transports like buses or subways which follow the fixed routes everyday, taxi drivers plan their own routes once they drop off a passenger. This is the main reason that different drivers get discrepant incomes. Figure 3 reveals some statistics w.r.t. 12,000 taxicabs during 110 days. As shown in Figure 3(a), the profit of a taxi driver can be measured by the fare (occupied) distance per unit working time, based on which, we divide the taxi drivers into 3 groups, the top 10% are regarded as high-profit drivers, the bottom 10% are considered as the low-profit drivers and the rests are medium part.

There is no doubt that at peak hours, the taxicabs are easy to find passengers. i.e., the taxis are often in short supply. However, at off-peak hours, the gap between the high-profit drivers and the low-profit drivers becomes obvious. Figure 3(b) further shows the time-variant occupied ratio (the quotient between the occupied distance and the whole distance) pertaining to the high/low-profit taxi drivers as well as the overall occupied ratio changing during a day. It's clear that from 10am to 3pm, the gap between the high-profit drivers and low-profit drivers is more significant. The critical factor determining the profit of a taxi driver depends on two folds. One is that the driver should know the places where he/she



(a) Distribution of profit (b) Occupied ratio during a day

Figure 3. Statistics on the profit distribution and occupied ratio

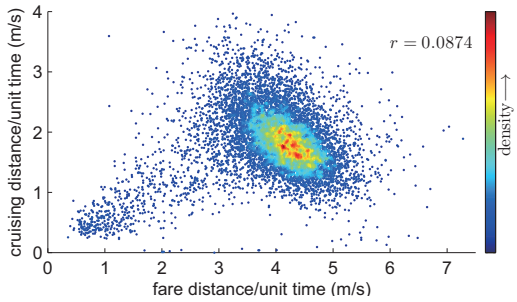


Figure 4. Density scatter of cruising distance/unit time w.r.t. profit

can pick up passengers quickly given a particular time of day. The other is the length of the typical trips that originate from a pick-up place. As we know, transportation terminals, shopping centers and hotels all generate demand for taxi service. A professional taxi driver usually knows when certain planes and trains arrive, when the movie is over at a local theater and even what time shifts change at certain businesses.

Typically, for an experienced local driver, instead of random cruising, they usually have a place to go with the intention to pick up new passengers after he/she drops off a passenger. Figure 4 presents an informative density scatter of the cruising distance per unit time w.r.t. the profit (measured by fare distance per unit time) for the time interval 10am to 3pm. The Pearson correlation coefficient of these two variables is only 0.0874 according to the plotted data. The color indicates the density of a point. This figure shows us that cruising more does not mean earning more. Instead, waiting at some right places may bring more chance to pick up a passenger. As shown in the figure, quite a few drivers cruise more than the majority (the points on the upper left corner of the hot kernel), however, their profit is lower. The right bottom parts (of the hot kernel) are the drivers who earn more but cruise less than the majority. We also conduct a survey among more than 10 local taxi drivers. According to their answers, after they drop off a passenger, 8 of them often have an intentionally nearby destination (the parking place we defined) to go, especially at off-peak hours. Based on their experience, rather than wasting gas when random cruising, they prefer to wait at a parking place with more chance to get a passenger.

Framework

The framework is illustrated in Figure 5. We develop an approach to detect the parking places from GPS trajectories and segment the GPS trajectories according to Definition 4, then map-match the GPS trajectories to road networks using the IVMM algorithm [13], which outperforms other approaches for low-sampling-rate GPS trajectories. Later, we utilize the detected parking places and the mapped trajectories to learn the time-dependent taxi behaviors. These processes are performed offline and will be repeated only when the trajectory data is updated. Leveraging the learned statistical results, we formulate a probabilistic model to integrate the taxi behaviors on each road segment and parking place as well as the mobility patterns of passengers. Based on this model, we perform real-time recommendations to maximize the profit of a taxi driver, and the possibility to get a vacant taxi for people respectively, given the location and time of a taxi driver/passenger.

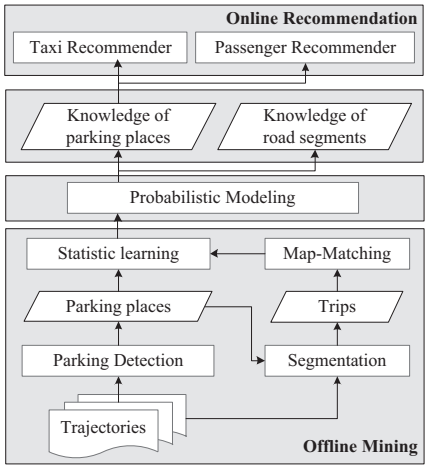


Figure 5. System Overview

MODEL DESCRIPTION

Taxi Recommender

The taxi recommender aims to provide the taxi drivers with the best parking places and the routes to these parking places. But how to define a "good" parking place? After a taxi drops off a passenger at time T_0, what the driver hopes is to find a new passenger as soon as possible. It would be best that the next trip is as long as possible, thus the driver can earn more money from the next trip. So a good parking place should bring a high probability to get a passenger, a short waiting time and a long distance of the next trip.

Assume P is a certain parking place and $R : r_1 \rightarrow r_2 \rightarrow \ldots \rightarrow r_n$ is a route to P. We say the driver *takes action* Λ_{RP} if he/she drives along R until finding a new passenger and waits at P for at most t_{max} time if he/she does not pick up a passenger along R. In this subsection, we answer the following questions:

1. How likely will the driver pick up a passenger if he/she takes the action Λ_{RP}?
2. If the driver takes action Λ_{RP} and succeeds in finding a new passenger, what is the expected duration from T_0 to the beginning of the next trip?
3. If the driver takes action Λ_{RP} and succeeds in finding a new passenger, how long is the expected distance/travel time of the next trip ?

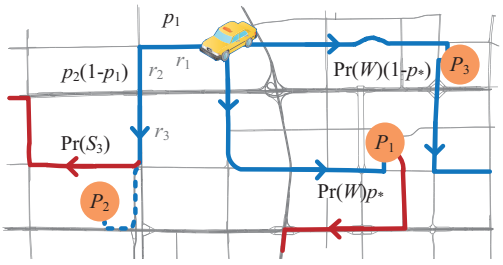


Figure 6. Taxi recommendation model

The Probability of Picking up the Next Passenger

Let S be the event that the driver *succeeds* in picking up the next passenger if he/she takes the action Λ_{RP} and $\overline{S}$ be the opposite situation (fails to get the next passenger). Then we have

$$S = \bigcup_{i=1}^{n+1} S_i, \tag{1}$$

where S_i for $i = 1, 2, \ldots, n$ is the event that the driver picks up a passenger at road segment r_i, and S_{n+1} is the event that the driver picks up a passenger at the parking place. Note that both S and S_i are with respect to the current time T_0. Let $t_i = \sum_{j=1}^{i} r_j.t$, i.e., the travel time from the start point to r_i. Denote the probability that a cruising taxi picks up a passenger at road segment r_i and at time $T_0 + t_i$ by

$$p_i = \Pr(\mathcal{C} \rightsquigarrow \mathcal{O} | r_i, T_0 + t_i). \tag{2}$$

Let

$$p_* = \Pr(\mathcal{P} \xrightarrow{(0, t_{max}]} \mathcal{O} | T_0 + t_n) \tag{3}$$

be the probability that a taxi succeeds in picking up a passenger at parking place P and waiting time $T_P \in (0, t_{max}]$ if the driver reaches P at $T_0 + t_n$. Then

$$\Pr(S_i) = \begin{cases} p_1, & i = 1 \\ p_i \prod_{j=1}^{i-1} (1 - p_j), & i = 2, 3, \ldots, n, \\ p_* \prod_{j=1}^{n} (1 - p_j), & i = n + 1. \end{cases} \tag{4}$$

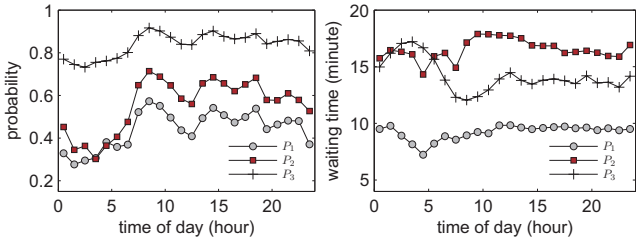
Now the answer of question 1 is clear:

$$\begin{aligned} \Pr(S) =& 1 - \Pr(\overline{\bigcup_{i=1}^{n+1} S_i}) \\ =& 1 - (1 - p_*) \prod_{j=1}^{n} (1 - p_j). \end{aligned} \tag{5}$$

The factor $\prod_{j=1}^{n}(1-p_j)$ in Equation 5 is the probability that the driver fails to find a passenger along R. We denote this event by $\overline{S_R}$. Note the route from the current position of the driver to P is not unique. Should we suggest the driver with the route that has the minimum $\Pr(\overline{S_R})$? It's obviously absurd since the driver can traverse all the road network in that case. In practice, we can provide the fastest route or a route with the minimum $\Pr(\overline{S_R})$ conditioned by that the distance does not exceed a threshold. This can be implemented by a simple generalization of the constrained shortest path problem [14].

Figure 7(a) plots the $\Pr(S)$ of three nearby parking places around Rear Sea (a bar district of Beijing). It's clear the probability fluctuates with time significantly. At peak hours (8-9am, 5-6pm) the probability is relatively higher than other time intervals for all these parking places.

Duration Before the Next Trip T



(a) $\Pr(S)$ (b) $E[T|S]$

Figure 7. $\Pr(S)$ **and** $E[T|S]$ **changing over time**

Let random variable T be the duration from current time T_0 to the beginning of the next trip, given that the taxi driver takes the action Λ_{RP}. Then T is a summation of two random variables: the cruising time along R, denoted as T_R and the waiting time at P, termed as T_P, i.e.,

$$T = T_R + T_P. \tag{6}$$

Note that T_R and T_P are not independent. Actually,

$$\begin{cases} T_P = 0, & \text{if } T_R \le t_n, \\ T_R = t_n, & \text{if } T_P > 0. \end{cases} \tag{7}$$

The probability mass function is given by

$$\begin{aligned} & \Pr(T_R = t_i | S) \\ =& \Pr(T_R = t_i, S) / \Pr(S) \\ =& \begin{cases} \Pr(S_i) / \Pr(S), & i = 1, 2, \ldots, n-1, \\ \dfrac{\Pr(S_n) + \Pr(S_{n+1})}{\Pr(S)}, & i = n, \end{cases} \end{aligned} \tag{8}$$

thus the conditional expectation of T_R is

$$\begin{aligned} \mathbf{E}[T_R | S] =& \sum_{i=1}^{n} t_i \Pr(T_R = t_i | S) \\ =& \frac{1}{\Pr(S)} \left(\sum_{i=1}^{n} t_i \Pr(S_i) + t_n \Pr(S_{n+1}) \right). \end{aligned} \tag{9}$$

Let W be the event that the driver waits at P, we have

$$\Pr(W) = \prod_{j=1}^{n} (1 - p_j). \tag{10}$$

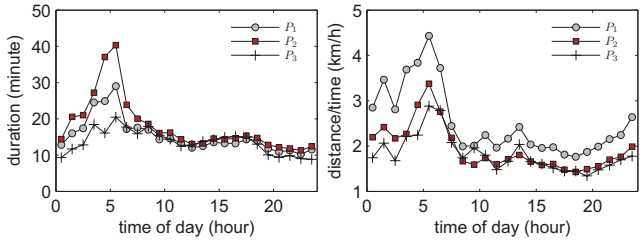
To learn the distribution, we break the interval $(0, t_{max}]$ into m buckets. Specifically, let

$$\begin{cases} t_0 = 0, \\ \triangle t^* = t_{max} / 2m, \\ t_j^* = (2j - 1) \triangle t^*, \ j = 1, 2, \ldots, m, \end{cases} \tag{11}$$

where t_j^* is the average waiting time for the j-th bucket. Denote the probability that the taxi succeeds in picking up a passenger and the waiting time T_P belongs to the j-th bucket by

$$p_*^j = \Pr(\mathcal{P} \xrightarrow{(t_j^* - \triangle t^*, t_j^* + \triangle t^*]} \mathcal{O} | T_P > 0, T_0 + t_n). \tag{12}$$

Actually, recall Equation 3, we have $p_* = \sum_{j=1}^{m} p_*^j$.



(a) $\mathbf{E}[T_N|S]$ (b) $\mathbf{E}[D_N|S]/E[T+T_N|S]$

Figure 8. Expected duration of next trip and expected profit w.r.t. time

The conditional probability

$$\Pr(T_p = t_j^*|S) = \begin{cases} \dfrac{(1-\Pr(W))}{\Pr(S)}, & j = 0, \\ \dfrac{\Pr(W)p_*^j}{\Pr(S)}, & 1 \le j \le m. \end{cases} \tag{13}$$

Therefore, the conditional expectation of T_P is

$$\mathbf{E}[T_P|S] = \frac{\Pr(W)}{\Pr(S)} \sum_{j=1}^{m} p_*^j t_j^*. \tag{14}$$

Then, the conditional expectation of T is

$$\begin{aligned} &\mathbf{E}[T|S] \\ =&\mathbf{E}[T_R|S] + \mathbf{E}[T_P|S] \\ =&\frac{\sum_{i=1}^{n} t_i \Pr(S_i) + t_n \Pr(S_{n+1}) + \Pr(W) \sum_{j=1}^{m} p_*^j t_j^*}{\Pr(S)}. \end{aligned} \tag{15}$$

As shown in Figure 7(b), the values of $\mathbf{E}[T]$ of the three parking places (using the same query point as Figure 7(a)) are depicted, changing over time smoothly. Note that the parking place P_1 has the shortest expected duration, yet has the lowest $\Pr(S)$, i.e., it is not so likely to pick up a passenger.

Distance/Travel Time of the Next Trip D_N, T_N

Let random variable D_N be the distance of the next trip if the driver takes the action Λ_{RP} conditioned by that S happens. Let q_i^j be the probability that the distance of the first trip satisfies $d_{j-1} < D_N \le d_j$, when S_i happens (note the time at that moment is $T_0 + t_i$), i.e., $\forall i = 1, 2, \ldots, n+1$,

$$q_i^j = \Pr(d_j - \triangle d < D_N \le d_j + \triangle d | S_i, T_0 + t_i). \tag{16}$$

Here,

$$\begin{cases} d_0 = 0, \\ \triangle d = d_{max}/2s, \\ d_j = (2j-1)\triangle d, \quad j = 1, 2, \ldots, s, \end{cases} \tag{17}$$

where d_{max} is the maximum distance of the first trip. Then, the conditional probability distribution is given by:

$$\Pr(D_N = d_j|S) = \sum_{i=1}^{n+1} \Pr(S_i) q_i^j / \Pr(S), \tag{18}$$

for $j = 1, 2, \ldots, s$. Thus, the conditional expected distance of the next trip is

$$\begin{aligned} \mathbf{E}[D_N|S] &= \frac{1}{\Pr(S)} \sum_{j=1}^{s} \left(d_j \sum_{i=1}^{n+1} \Pr(S_i) q_i^j \right) \\ &= \frac{1}{\Pr(S)} \sum_{i=1}^{n+1} \Pr(S_i) \left(\sum_{j=1}^{s} d_j q_i^j \right). \end{aligned} \tag{19}$$

Note that the conditional expected travel time of the next trip $\mathbf{E}[T_N|S]$ is computed in exactly the same way as $\mathbf{E}[D_N|S]$, thus we omit the detail. Figure 8 plots the expected duration and distance of the next trip, w.r.t. time of day. As we stated above, this area is the bar district. People mainly come to this area at night and stay until the dawn of the next day. Mostly, people who go to this place live not so close to this area, thus both the expected duration and distance of the next trip at pre-dawn period is higher than other time of day.

Passenger Recommender

Different from the taxis, the passengers do not want to walk too long for hailing a taxi. If a passenger is close to at least one parking place, we suggest him to go to the nearest parking place. Otherwise, we suggest the passenger with the most possible road segments nearby on which they can find a vacant taxi. This is much easier than the taxi recommendation problem. Let $\Pr(\mathcal{C}; r|t)$ be the probability that there is a vacant taxi on road segment r at time t, given the passenger's current position, we suggest the road segments which have the highest $\Pr(\mathcal{C}; r|t)$ among a reachable region Ω of the passenger, i.e.,

$$r = \operatorname*{argmax}_{r \in \Omega} \Pr(\mathcal{C}; r|t). \tag{20}$$

Later, we discuss in detail how to learn the needed probabilities proposed in this section.

OFF-LINE MINING

Parking Places Detection

This section details the process for detecting parking status from a non-occupied trip and accordingly finding out the parking places in the urban area of a city based on a collection of taxi trajectories.

Candidates Detection

Figure 9 demonstrates the parking candidate detection approach, given a non-occupied trip $p_1 \to p_2 \to \cdots \to p_7$. We

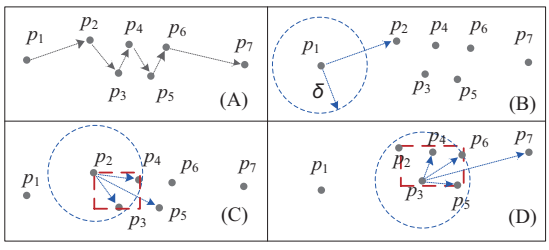


Figure 9. Parking candidates detection

Algorithm 1: ParkingCandidateDetection

Input: A road network G, a trajectory Tr, distance threshold δ, time threshold τ
Output: A set of parking candidates $\mathbb{P} = \{P\}$

```
1  i ← 0, M ← ||Tr||, P ← ∅, ℙ ← ∅;
2  while i < (M − 1) do
3      j ← i + 1; flag ←false;
4      while j < M do
5          dist ←Distance (p_i, p_j);
6          if dist < δ then j ← j + 1;flag =true;
7          else break;
8      if p_{j−1}.t − p_i.t > τ and flag =true then
9          foreach point p ∈ Tr[i, j) and p ∉ P do
10             P.Add(p);/* build a candidate */
11         if i = j − 1 then
12             ℙ.Add(MB (P)); P ← ∅;
               /* add the minimum bounding box of P
                  into ℙ */
13         i ← i + 1;
14     return ℙ
```

first keep on checking the distance between the current point and the latter point until the distance is smaller than a threshold. As depicted in Figure 9 B), since $dist(p_1, p_2)$ exceeds the distance threshold δ, we move next, fixing p_2 as the "pivot" point and find that $dist(p_2, p_3) < \delta, dist(p_2, p_4) < \delta$ while $dist(p_2, p_5) > \delta$ (Figure 9 C). If the time interval between $p_2.t$ and $p_4.t$ is larger than the time threshold τ, the three points form a small cluster represent a possible parking candidate. Next, we fix p_3 as the pivot point and keep on the procedure to check latter points. Finally, as shown in Figure 9 D), we detect $(p_2, p_3, p_4, p_5, p_6)$ as a parking candidate because we cannot expand this group any further, i.e., all the points in this group have a distance farther than δ to p_7. The pseudocode is provided in Algorithm 1.

Filtering

Essentially, the candidate detection algorithm finds out the locations where the GPS points of a taxi are densely clustered, with spatial and temporal constraints. However, a parking candidate could sometimes be generated by taxis stuck in a traffic jam, or waiting for signals at a traffic light, instead of a real parking. To reduce such false selections, we design a supervised model for picking out the true parking status from the candidate sets, using the following features:

- *Spatial-Temporal features* including 1) Minimum Bounding Ratio (MBR). As shown in Figure 10(A),B)), MBR is the area ratio between the bounding box of the road segment (MBRr) and the bounding box of the GPS points (MBRc) in the candidate set. 2) AverageDistance. The average distance d_c between points in the candidate set and their nearest road segments, as shown in Figure 10 C). 3) CenterDistance. The distance between center point in MBRc of the candidate set and the road segments. 4) Duration. The parking duration of a candidate. 5) LastSpeed. The speed of the last point leaving a parking candidate.
- *POI feature*. As we know, a parking place is highly relevant with the points of interests (POI) around it, e.g., subway exits, theaters, shopping malls within 50 meters, shown in Figure 10 C). We employ the *term frequency-inverse document frequency* (tf-idf)[10] to measure the importance of a POI to a parking place. Specifically, for a given parking place, we formulate a POI vector, $\langle v_1, v_2, \ldots, v_k \rangle$ where v_i is the tf-idf value of the i-th POI category, given by:

$$v_i = \frac{n_i}{N} \times \log \frac{\|\mathbb{P}\|}{\|\{P \in \mathbb{P} | \text{the } i\text{-th POI category} \in P\}\|},$$

where n_i is the number of POIs belonging to the i-th category, N is the number of POIs lying around the parking candidate. The idf item is calculated using the quotient of the number of parking candidates divided by the number of parking candidates which have the i-th POI category, and then taking the logarithm of that quotient.
- *Collaborative feature*. For a real parking place, other drivers should also park historically at that place. Otherwise, it's not a common parking place. So we use the number of parking candidates within 50 meters in the past 7 days of a candidate as a collaborative feature to enhance the classifier.

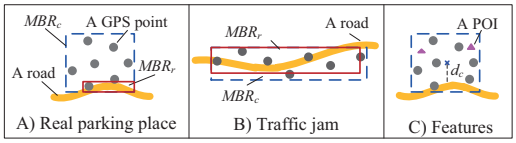


Figure 10. Parking candidates filtering

We use a human-labeled dataset to learn the threshold and train a bagging [2] classifier model to guarantee the high precision and recall of the detected parking candidates (The results will be shown later in the evaluation part). Then we utilize the model to inference whether a candidate is a really parking or a traffic jam.

Parking Place Clustering

The parking status is detected for each trajectory separately. However, the parking place detected from a single trajectory is only a portion of a real parking place. Thus different parking places may be actually the same one. We use a density based clustering method OPTICS [1] aiming to discover the essentially same parking places. The reason for using this method is that it outperforms other methods when the clustered region may have an arbitrary shape and the points inside a region may be arbitrarily distributed. As shown in Figure 11, the left figure plots all the parking candidates in the area of Beijing West Railway Station and the right one shows the results after filtering and clustering, with the color indicating the number of candidates in a cluster.

Learning the Time-dependent Probabilities

In practice, we assume the probability is stable during time interval $[t, t + \triangle t]$, where $\triangle t$ is a fixed threshold. This is reasonable, since the probability changes gradually instead of sharply. For computing the time-dependent probability, a common way is to partition a day into fixed slots (e.g., one hour a slot), and calculate the result for each slot beforehand. Different from this method, we develop a *partition-and-group* approach so as to compute this probability "just-in-time" and enable real-time recommendation. More con-

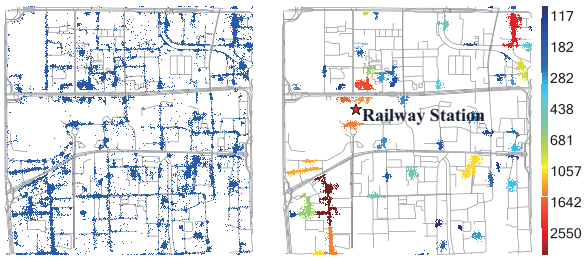


Figure 11. Candidate parking places (left) and clustered parking places (right) in the area of Beijing West Railway Station. The color indicates the number of parking candidates detected for each cluster.

cretely, we partition one day into K small time intervals, the length of each interval is τ (where $\triangle t$ can be divisible by τ, e.g., $\tau = 5$ minutes, $\triangle t = 30$ minutes). Thus, the k_{th} interval is

$$I_k = [(k-1)\tau, k\tau],\ k = 1, 2, \ldots, K. \tag{21}$$

Then we learn the statistical result for each I_k offline. In the online phase, when the time t of a taxi driver/passenger is input, we retrieve the corresponding intervals (i.e., a set of I_k) which belongs to $[t, t+\triangle t]$, then compute the corresponding probability using the statistical results obtained from all the retrieved intervals. The intuition of this partition-and-group approach is much like the Riemann Integral. The advantage of the above "just-in-time" way compared to the fixed slot method is that we can avoid the discontinuity when crossing the boundary of a interval (e.g., the probability at 10:59 am may entirely different with the one at 11:00am if 11:00am is the boundary of a fixed time slot) as well as make the most use of the sparse data in a small interval.

The Probabilities w.r.t. Road Segments

For computing the probability on a road segment, we need to detect all the state transitions. Due to the *low sampling rate* [13] problem, the exact point that the transition occurs may not be observed from the data. In this situation, we adopt the convention that the transition happens at the road segment on which the previous state is observed, i.e., if

$$\begin{cases} p_i.s \neq p_{i+1}.s, \\ p_i.r \neq p_{i+1}.r, \end{cases} \tag{22}$$

we insert a p_i' between p_i and p_{i+1} on $p_i.r$, with state $p_{i+1}.s$.

Instead of computing the probability on each road segment, we first conduct a road segment clustering to integrate the road segments with similar features so as to tackle the data sparseness problem and accelerate the online computing. We identify the following features (as input for road segment clustering), which are derived from the underlying road structure and POIs.

- L: The actual length of a road segment.
- L/E: The ratio between L and the Euclidean length (between the terminal points) E of a road segment. The larger the value is, the more tortuous the road segment is.
- *dir*: The direction of a road segment (one-way/two-way).
- *Lanes*: The number of lanes in a given road segment.
- *degree*: The in/out-degree of a given road segment.
- *POI*: The POI feature is defined similarly with the parking place detection.

As a result, we obtain a collection of clusters, each of which contains a set of road segments with similar features. Then the statistical learning is performed in terms of each cluster. Let $\widetilde{r}$ be the cluster road segment r belongs to, and $\widetilde{R}$ be the set of all the clusters. Let $\#_k(\mathcal{C}; \widetilde{r})$ be the number of trips that the taxis once have been at the $\mathcal{C}$ state during I_k on all the road segments within cluster $\widetilde{r}$ versus $\#_k(\mathcal{O}; \widetilde{r})$ for the occupied state. Then the probability that there exist a taxi cruising on r at time t is computed by:

$$\Pr(\mathcal{C}; r|t) = \frac{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor (t+\triangle t)/\tau \rfloor} \#_k(\mathcal{C}; \widetilde{r})}{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor (t+\triangle t)/\tau \rfloor} \sum_{\widetilde{r} \in \widetilde{R}} (\#_k(\mathcal{C}; \widetilde{r}) + \#_k(\mathcal{O}; \widetilde{r}))}. \tag{23}$$

The probability $\Pr(\mathcal{O}; r|t)$ can be similarly computed.

Let $\#_k(\mathcal{C} \rightsquigarrow \mathcal{O}; \widetilde{r})$ be the number of trips that the taxis transfer from the cruising state to occupied state, i.e., pick up a passenger during I_k when cruising on all road segment within cluster $\widetilde{r}$, then

$$\Pr(\mathcal{C} \rightsquigarrow \mathcal{O}|r, t) = \frac{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor (t+\triangle t)/\tau \rfloor} \#_k(\mathcal{C} \rightsquigarrow \mathcal{O}; \widetilde{r})}{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor (t+\triangle t)/\tau \rfloor} (\#_k(\mathcal{C}; \widetilde{r}))}. \tag{24}$$

With regard to the distance D_N,

$$\Pr(d_a < D_N \leq d_b|r, t) = \frac{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor t+\triangle t/\tau \rfloor} \#_k(d_a, d_b; \widetilde{r})}{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor (t+\triangle t)/\tau \rfloor} \#_k(0, d_{max}; \widetilde{r})}, \tag{25}$$

where d_{max} is the maximum distance of a trip and $\Pr(t_a < T_N \leq t_b \mid r, t)$ is similarly computed.

The Probabilities w.r.t. the Parking Places

For each cluster of the parking places, we have a set of trajectories which are at the parking state with varied arriving time and leaving time. Let $\#_k(t_a, t_b, \mathcal{P} \rightsquigarrow \mathcal{O}; P)$ be the number of trips that start from parking place P when the taxi driver arrives at P during I_k and finally become occupied with the waiting time $T_P \in (t_a, t_b]$. Let $\#_k(\mathcal{P} \rightsquigarrow \mathcal{O}; P)$ be the number of trips that originate from P after the driver arrives at P during I_k and becomes occupied when leaving P versus $\#_k(\mathcal{P} \rightsquigarrow \mathcal{C}; P)$ denotes the taxis which are still non-occupied (cruising) when leaving P. Then the probability that the waiting time $T_P \in (t_a, t_b]$ when reaching P at t can be calculated by

$$\Pr(\mathcal{P} \xrightarrow{(t_a, t_b]} \mathcal{O}|T_P > 0, t) = \frac{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor t+\triangle t/\tau \rfloor} \#_k(t_a, t_b, \mathcal{P} \rightsquigarrow \mathcal{O}; P)}{\sum_{k=\lfloor t/\tau \rfloor}^{\lfloor (t+\triangle t)/\tau \rfloor} (\#_k(\mathcal{P} \rightsquigarrow \mathcal{O}; P) + \#_k(\mathcal{P} \rightsquigarrow \mathcal{C}; P))}. \tag{26}$$

ONLINE RECOMMENDATION

In this stage, given the location and time of a taxi driver/passenger, we provide real-time recommendation based on the proposed probabilistic model and the derived statistical knowledge.

For the taxi recommender, we first perform a range query according to the location of the taxi, and then retrieve a set of potential parking places. For each parking place P, we generate the route R with the minimum $\Pr(\overline{S_R})$ using a dynamic programming recursion [14] in parallel. Then we compute the probability $\Pr(S)$ and the conditional expectations: $\mathbf{E}[T|S]$, $\mathbf{E}[D_N|S]$, $\mathbf{E}[T_N|S]$ according to the query time. Later, we rank the candidate parking places with (but not limited to) the following strategies (S1–S3) and accordingly recommend top-k parking places to the driver in real-time. The thresholds P_θ, D_θ and F_θ can either be learned from the training data or be set by the user.

S1. $Topk_{max}\{\mathbf{E}[D_N|S]/\mathbf{E}[T+T_N|S] : \Pr(S) > P_\theta\}$. The candidate parking places of this strategy are restricted to the ones which have a $\Pr(S)$ larger than a threshold P_θ, among which, we provide the taxi driver with the top-k profitable parking places, i.e., the driver can earn the most money per unit time by traveling to these k parking places.

S2. $Topk_{min}\{\mathbf{E}[T|S] : \Pr(S) > P_\theta, D_N > D_\theta\}$. This strategy retrieves k parking places which have the minimum expected duration before picking up a new passenger and have at least P_θ possibility to pick up a passenger as well as D_θ distance of the next trip.

S3. $Topk_{max}\{\Pr(S) : \mathbf{E}[D_N|S]/\mathbf{E}[T+T_N|S] > F_\theta\}$. This strategy provides the parking places, towards which the drivers are most likely to pick up a passenger and has a guaranteed profit (at least F_θ).

For the passenger recommender, we also perform a range query so as to obtain a region, which is within a walking distance of the passenger. If this region contains parking places, we suggest the passenger with the k nearest parking places. Otherwise, according to Equation 20, we return the road segments with k largest probability of having a vacant taxi.

VALIDATION

Settings

Dataset

Road network: We evaluate our method using the road network of Beijing, which contains 106,579 road nodes and 141,380 road segments.

Trajectory: The dataset contains the GPS trajectory recorded by over 12,000 taxis in a period of 110 days. The total distance of the data set is more than 200 million kilometers and the number of points reaches to 577 million. After trip segmentation, there are in total 20 million trips, among which 46% are occupied trips and 53% are non-occupied trips. We use 70 days' (random selected) data to build our system and evaluate the method using the rest (40 days) data.

Evaluation on Parking Place Detection

We first evaluate the effectiveness of filtering, i.e., whether our method can identify a taxi is parking or is stuck in a traffic jam. We ask three local people to label 1000 parking candidates (True/False). The precision and recall w.r.t. the features we used for the classifier is presented in Table 2. As a result, we get a 91% precision and 89% recall which is enough for detecting the true parking places and clustering.

Features	Precision	Recall
Spatial	0.695	0.670
Spatial+POI	0.716	0.696
Spatial+POI+Collaborative	0.725	0.706
Spatial+POI+Collaborative+Temporal	0.909	0.889

Table 2. Results of parking place filtering

We evaluate the performance of parking place clustering by two methods: 1) We conduct a survey towards more than 20 users and ask them for submissions of parking places they know/have seen. We received over 70 parking places which basically uniformly distributed in the urban area of Beijing. Then we use this labeled data to test the recall of parking places generated by our parking place clustering approach. The recall of the labeled parking places reaches to 81%, i.e., 81% of the labeled parking places are involved in the clustered parking places. 2) We check the ratio of positive instances compared with all the parking candidates (using the test set) for each clustered parking place. The mean value of the ratio is around 89%.

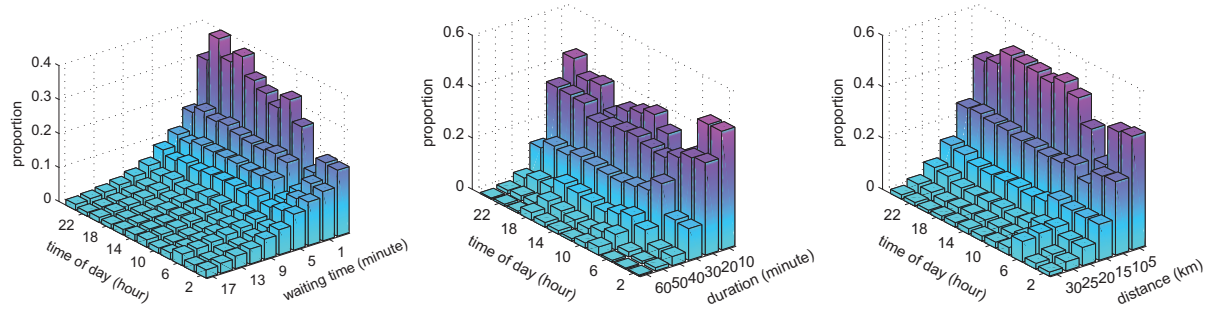
Evaluation on Statistical Learning

Based on our model, we calculate the overall time-dependent distribution for both the parking places (Figure 12) and the road segments (Figure 13). For example, as shown in Figure 12(a), the average waiting time T_P are mostly less than 10 minutes. During the midnight, the distribution of waiting time is comparative decentralized while after 8am, the waiting time trends to be shorter. That means, the driver can get a passenger with a shorter waiting time than in midnight, which is quite accord with the common sense. Figure 13(a) depicts the average probability (for each level of road segments) that a taxi transfer from the cruising status to the occupied status changing over time. Since the level-0 roads and level-1 roads are mainly high-ways or main roads, the probability is reasonable lower compared with level-2/3.

Evaluation on Online Recommendation

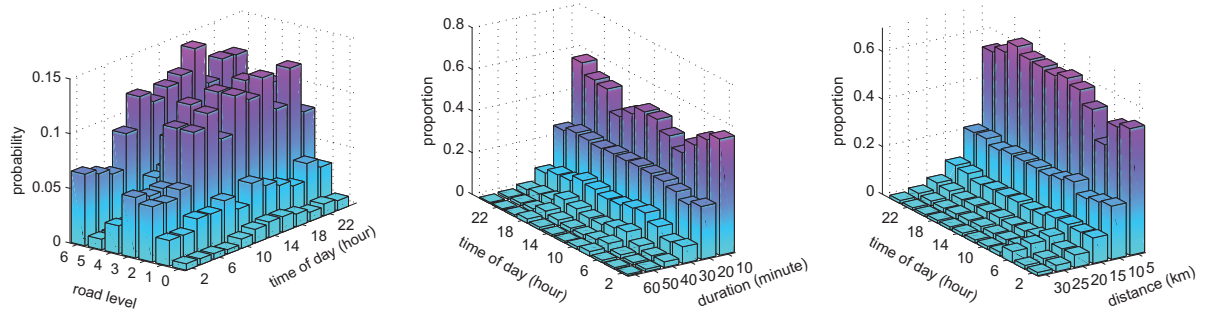
For preprocessing the test set, we extract the trajectories of the detected high profit drivers and segment them to occupied/cruising/parking trips. Before each parking or occupied trip, we randomly select 10 points which are less than 3km faraway (which is the radius of our range query for retrieving the parking places) to the next pick-up point/parking place as the query points. For each query point, the groundtruth contains the following information: query ID, query time, geo-position, the routes before next pick-up point, and the parking places they've gone to before next pick-up point (ordered by timestamps). Then we evaluate the performance of our method using three ranking strategies (S1–S3) as well as two baselines: B1) suggest the top-k nearest parking places and B2) recommend the top-k parking places which have the largest overall probability of picking up a passenger). We measure the effectiveness of these methods using three criterion:

1) *Precision and Recall*. The precision is the ratio between number of hits and the number of recommendations. The recall measures the fraction of the parking places the drivers actually go to that are suggested.



(a) waiting time (b) duration of the first trip (c) distance of the first trip

Figure 12. Distribution in parking places (overall)



(a) $\Pr(\mathcal{C} \rightsquigarrow \mathcal{O})$ (b) duration of the first trip (c) distance of the first trip

Figure 13. Statistics results of road segments (overall)

Rule	Condition	Score
(i)	$R_i = R_1^*$ and the driver picks up a passenger on R_1^*	3
(ii)	$R_i = R_1^*$ and $P_i = P_1^*$	3
(iii)	(i),(ii) do not hold, but $R_i = R_1^*$ or $P_i = P_1^*$	2
(iv)	(i),(ii),(iii) do not hold, but $\exists j$ s.t., $R_i = R_j^*$ or $P_i = P_j^*$	1
(v)	none of the above holds	0

Table 3. Scoring rules

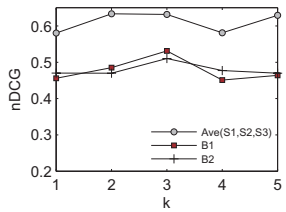


Figure 14. nDCG

	S1	S2	S3	B1	B2
Precision	0.63	0.66	0.67	0.60	0.61
Recall	0.59	0.65	0.64	0.57	0.52

$RME(T)$	0.15
$RME(D_N)$	0.02
$RME(T_N)$	0.03

Table 4. RME, precision and recall

2) *Normalized Discounted Cumulative Gain at the k-th position [4] ($nDCG_k$).* $nDCG_k = \frac{DCG_k}{IDCG_k}$, where $DCG_k = \sum_{i=1}^{k} \frac{S(i)}{\log(1+i)}$ and $IDCG_k$ denotes the DCG_k value for an ideal ranking, given $S(i)$ is the scoring function for the i-th recommendation. Given a driver's position and time, we recommend him/her with top-k parking places and the corresponding routes. Let (R_j^*, P_j^*), $j = 1, 2, \ldots, m$ be the routes the driver actually traversed, and the parking places the driver waited at, before he/she picked up the next passenger (note that the driver may wait at several parking places or didn't wait at any parking place). Assume the top-k parking places we recommend to a driver are denoted by (R_i, P_i), $i = 1, 2, \ldots, k$. The scoring function for (R_i, P_i) is determined according to Table 3. For routes R_i and R_j^*, we consider them to be the same if there is a significant overlap (90%, say) between R_i and R_j^*. Then we compute the $nDCG_k$ for each query point and take the average $nDCG_k$ among all the queries as the overall $nDCG_k$.

3) For the hit parking places (the driver go to the suggested parking places and pick up a passenger finally), we further study the precision of the predicted value, i.e., T, D_N and T_N measured by *Relative Mean Error* (RME), e.g., given the real duration before the next trip T^* , then

$$RME[T] = \mathbf{E}\left[(T^* - \mathbf{E}\left[T|S\right])/T^*\right].$$

Figure 14 plots the nDCG changing over k, we visualize the average nDCG value of S1, S2, and S3 as the nDCG for our method. Overall, our approach has a 0.1 improvement than the competing methods in terms of nDCG_2. Table 4 presents the other results obtained from the evaluation. The S3 strategy has the best performance in terms of precision and S2 is the best for the recall. The error, measured by *RME*, is less than 3% for D_N and T_N and 15% for T.

RELATED WORK

Dispatching Systems

Taxi dispatching systems are attracting growing attention from researchers with the development of intelligent transportation systems and the popularization of GPS sensors [5]. Most existing dispatching systems assign a task to taxi drivers based on nearest neighbor principle in terms of distance or time. Phithakkitnukoon et al. [8] use the naive Bayesian classifier with developed error-based learning approach to infer the

number of vacant taxis at a given time and location which can be used to enhance the dispatching system. Yamamoto et al. [11] propose a fuzzy clustering and adaptive routing approach to improve dispatching system by assigning vacant taxis adaptively to the locations with high expectation of potential customers.

Different from the centralized dispatching, our recommendation system provide suggestions to taxi drivers/passengers and let them make their own decisions at a road segment level (not a region or grid). Typically, for a dispatching system, the customers need to book a taxi by telephone/internet in advance, and it is usually not free of charge. Most passengers hail a taxi along the road or stand where available instead of booking a taxi. Besides, our method aims to maximize the profit for each taxi driver instead of balancing the income of all the taxi drivers which is usually a goal of a dispatching system. In addition, our approach can be combined with a dispatching system so as to complement each other.

Location Recommendation For Taxi Drivers

Ge et al. [3] present a model to recommend a taxi driver with a sequence of pick-up points so as to maximize a taxi driver's profit. This work formulates the target problem by a mobile sequential recommendation (MSR) problem. Li et al. [6] study the passenger-finding strategies (hunting/waiting) of taxi drivers in Hangzhou. In this work, L1-Norm SVM is used to select features for classifying the passenger-finding strategies in terms of performance. Recently, Powell et al. [9] propose an approach to suggest profit (grid-based) locations for taxi drivers by constructing a Spatio-Temporal Profitability map, on which, the nearby regions of the driver are scored according to the potential profit calculated by the historical data.

Our approach is different from the above methods in the following aspects: 1) We provide recommendations to both taxi drivers and passengers, which mobilizes them and reduces the disequilibrium of the demand and supply. 2) Instead of a grid/cell-based partition of the map, our recommendation is provided on road-segment level, which enables more accurate and meaningful understanding of the taxi drivers' behaviors as well as a more practical recommendation for both the taxi drivers and the passengers. 3) We focus on the off-peak hours to help the driver make the first step decision whenever and wherever they want to decide a destination to go. In practice, the "first step" recommendation would be more effective since usually the drivers are not willing to remember a sequence of places. 4) We develop an algorithm to distinguish the parking status from traffic jams and propose a solution to detect the parking places in an urban area. 5) We target the challenges when building the system based on sparse data and facilitate the on-line recommendation with a partition-and-group framework.

CONCLUSION

Leveraging the pick-up behaviors learned from the high-profit taxi drivers and the mobility patterns of passengers, we build a recommendation system for both the taxi drivers and passengers. In this paper, motivated by the behaviors of discovered high-profit taxi drivers, an elaborate probabilistic model is devised to maximize the profit of a taxi driver and the possibility to find a vacant taxi for a passenger. We evaluate our method using a large number of GPS trajectories of taxicabs. The results show that our method can effectively provide the taxi drivers with high-profit locations, e.g., the $nDCG_2$ of our method has a 10% improvement than the baseline methods and the precision of the recommendation reaches to 67%. By mobilizing the drivers and passengers, our system can ease the supply/denamd disequilibrium problem to a certain extent. Furthermore, this recommender reduces the cruising distance of a taxi driver such that saves energy consumption and lighten the transportation pressure of a city.

In the future, we will incorporate the real-time traffic information to provide better routes towards the parking places (with high probability to pick up a passenger along a route and within a short travel time).

REFERENCES

1. M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *SIGMOD 1999*, pages 49–60. ACM Press, 1999.
2. L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
3. Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani. An energy-efficient mobile recommender system. In *Proc. KDD 2010*, pages 899–908.
4. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
5. D. Lee, H. Wang, R. Cheu, and S. Teo. Taxi dispatch system based on current demands and real-time traffic conditions. *Transportation Research Record: Journal of the Transportation Research Board*, 1882(-1):193–200, 2004.
6. B. Li, D. Zhang, L. Sun, C. Chen, S. Li, G. Qi, and Q. 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, march 2011.
7. New York City Taxi and Limousine Commission. Taxi of Tomorrow Survey Results, Feb 2011.
8. S. Phithakkitnukoon, M. Veloso, C. Bento, A. Biderman, and C. Ratti. Taxi-aware map: Identifying and predicting vacant taxis in the city. In *Proc. AMI 2010*, page 86.
9. J. Powell, Y. Huang, F. Bastani, and M. Ji. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *Proceedings of the 12th International Symposium on Advances in Spatial and Temporal Databases*, SSTD '11, 2011.
10. H. Wu, R. Luk, K. Wong, and K. Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–37, 2008.
11. K. Yamamoto, K. Uesugi, and T. Watanabe. Adaptive routing of cruising taxis by mutual exchange of pathways. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 559–566. Springer, 2010.
12. J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 99–108, New York, NY, USA, 2010. ACM.
13. J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Sun. An interactive-voting based map matching algorithm. In *Proc. MDM 2010*, pages 43–52.
14. M. Ziegelmann. *Constrained Shortest Paths and Related Problems*. PhD thesis, Universität des Saarlandes, 2001.