Ridesharing Recommendation: Whether and Where Should I Wait?

Chengcheng Dai(✉)

Department of Computer Science, City University of Hong Kong,
Kowloon Tong, Hong Kong
chengcdai2-c@my.cityu.edu.hk

Abstract. Ridesharing brings significant social and environmental benefits, e.g., saving energy consumption and satisfying people's commute demand. In this paper, we propose a recommendation framework to predict and recommend whether and where should the users wait to rideshare. In the framework, we utilize a large-scale GPS data set generated by over 7,000 taxis in a period of one month in Nanjing, China to model the arrival patterns of occupied taxis from different sources. The underlying road network is first grouped into a number of road clusters. GPS data are categorized to different clusters according to where their sources are located. Then we use a kernel density estimation approach to personalize the arrival pattern of taxis departing from each cluster rather than a universal distribution for all clusters. Given a query, we compute the potential of ridesharing and where should the user wait by investigating the probabilities of possible destinations based on ridesharing requirements. Users are recommended to take a taxi directly if the potential to rideshare with others is not high enough. Experimental results show that the accuracy about whether ridesharing or not and the ridesharing successful ratio are respectively about 3 times and at most 40 % better than the naive "stay-as-where-you-are" strategy. This shows that about 500 users can save 4–8 min with our recommendation. Given 9 RMB as the starting taxi fare and suppose users can save half of the total fare by ridesharing, users can save 10.828-44.062 RMB.

1 Introduction

Due to the emergency of saving energy consumption and assuaging traffic congestion while satisfying people's needs in commute and willings to save money in ride, ridesharing enabled by low cost geo-location devices, smartphones, social networks and wireless networks has recently received a lot of attention [1–4].

Taxis are considered as a major means of transportation in modern cities. In many big cities, taxis are equipped with GPS sensors to report their locations, speed, direction and occupation periodically. In Fig. 1 we gather statistics of the pick-up actions from a large-scale GPS data set generated by over 7,000 taxis in a period of one month in Nanjing, China. There are roughly 3 peaks per day where the pick-up number is above 6,000. We partition Nanjing metropolitan area into

B. Cui et al. (Eds.): WAIM 2016, Part I, LNCS 9658, pp. 151–163, 2016.
DOI: 10.1007/978-3-319-39937-9_12

30 × 30 grids with equal intervals. We count all the pick-up and drop-off events in 9:30am–9:35am on June 25th in each region and analysis the hotspots. As shown in Fig. 2a and b, both pick-up and drop-off have hotspots, indicating that many queries are likely to get ridesharing. Ridesharing gives the potential to solve congestion, pollution and environmental problems as well as saves money for users.

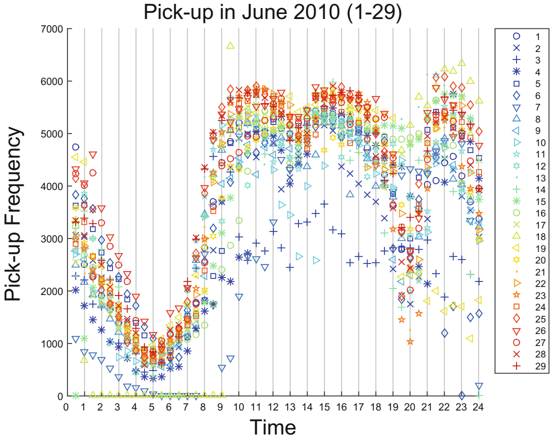


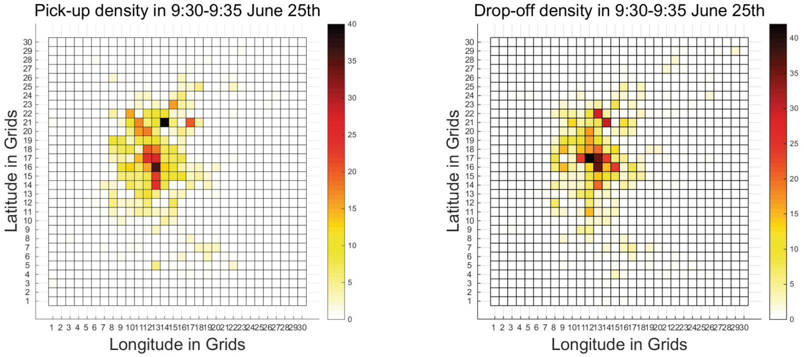
Fig. 1. Pick-up frequency in 30 min of Nanjing.

Ridesharing can be classified into carpooling [5], real-time taxi ridesharing [1,2,6], slugging [3] and Dial-a-Ride [7]. In slugging, passengers change their sources and destinations to join the trips of drives while drivers of others change their routes to pick up and drop off passengers. In real-time taxi ridesharing, ridesharing becomes an optimization problem to allocate a query to proper taxis considering extra cost in terms of both distance and waiting time. In Carpooling or recurring ridesharing, the driver is associated with her own trip. In Dial-a-Ride vehicles need to return to the same location (depot) after the trip.

Previous works on ridesharing mostly focus on the driver side by considering which taxi should the coming request be assigned to for the minimum extra travel time or travel distances. On the other hand, we focus on the user side of ridesharing to help the users decide whether they can rideshare and where should they wait if they are likely to get ridesharing. We use slugging as the ridesharing type since slugging is shown to be effective in reducing vehicle travel distance as a form of ridesharing [3].

Consider the scenario of slugging, Alice raises a ridesharing query $Q = (id, timestamp, l_s, l_d, t_s, t_e, t_w)$ where id is user id, $timestamp$ is when the query is submitted, l_s and l_d are respectively the source and the destination of the user, t_s is the maximum walking time to a new place, and t_e is the maximum walking time after she left the taxi to her own destination. t_w is the maximum waiting time at the new place for ridesharing. At present there is no taxi to rideshare

Alice in the system. Alice may either take a taxi directly or try again after a short time period. With the observation that Alice may also walk to some place nearby during waiting in order to increase her chance to rideshare, we propose a framework to predict the probability for Alice to rideshare and recommend whether and where should she wait. The key challenges are (i) how to embed query satisfaction requirements with recommendation, which makes our problem more complicated than finding passengers or taxis [8–10]; (ii) how to develop effective machine learning algorithms for ridesharing recommendation.



(a) density of pick-up

(b) density of drop-off

Fig. 2. Hotspots for pick-ups and drop-offs.

In the framework, ridesharing recommendation is based on the probabilities to have other passengers departing from somewhere within time t_s from source l_s towards somewhere within time t_e from destination l_e. Since taxi appearance is too dynamic for a single road, the underlying road network is grouped into road clusters. For each road cluster that are within time t_s from source l_s, we investigate the probability for taxis to depart from somewhere in it and have somewhere that is within time t_e to destination l_d as their destination by kernel density estimation [11]. Only road clusters with the probabilities greater than a threshold are recommended to users. In case many road clusters satisfies the condition, we return top-k clusters according the probabilities. If no cluster is returned for Q, the user will be suggested not to wait for ridesharing and take a taxi directly. Thus users can either save time or save money.

Rather than calculating a common distribution for all road clusters [12,13] to describe possible taxi destinations, we derive unique distributions for each road cluster. Given any new location (lon, lat), we can obtain the probability to have a taxi that departs from a certain cluster towards location (lon, lat). In reality, the probabilistic distributions should be various since clusters have different features such as Point-of-Interests (POIs). For instance, at noon taxi passengers from clusters with POIs of companies are likely to have restaurants as destinations while passengers from clusters with POIs of residencies may have companies as destinations.

To the best of our knowledge, this is the first work to study ridesharing from the aspect of predicting whether and where should a user wait. The main contributions of this paper can be summarized as follows:

- We design a recommendation framework to help users decide whether and where to wait for ridesharing.
- We explore the arrival patterns of road clusters based on kernel density estimation for ridesharing recommendation.
- Experimental results based on real GPS data set show that the accuracy about whether ridesharing or not and the ridesharing successful ratio are respectively about 3 times and at most 40 % better than the naive "stay-as-where-you-are" strategy, i.e., users wait at their sources for ridesharing. About 500 users can save 4–8 min with our recommendation. Given 9 RMB as the starting taxi fare and suppose users can save half of the total fare by ridesharing, users can save 10.828-44.062 RMB.

The rest of this paper is organized as follows. Section 2 highlights related works. Section 3 delineates the proposed recommendation framework. Section 4 analyzes the performance. Finally Sect. 5 concludes this paper.

2 Related Works

In this section, we highlight related works in both ridesharing and recommender systems based on taxi GPS data set.

2.1 Ridesharing

Ridesharing is transformed to an optimization problem about matching one driver and multiple ridesharing queries considering extra cost in terms of both distance and waiting time. We summarize the related works as dynamic matching and other issues like fair payment mechanism.

Dynamic Ridesharing Matching. Current dynamic ridesharing can be classified into four types, namely slugging, taxi ridesharing, carpooling and Dial-a-Ride. In slugging passengers change their sources and destinations to join the trips of drivers. From the passenger's point of view, this requires the source and destination of driver to be close to those of the passenger. In this paper we adapted slugging as our ridesharing type. The closeness is controlled by t_s and t_e in the query Q. With the walking speed, we can easily get the maximum walking distances of each query. In the other three types drivers change their routes to pick up and drop off passengers. Carpooling [5] is ridesharing based on private cars where the driver is associated with her own trip. Carpooling considers computing the best route for a given set of requests. In contrast to carpooling, taxi ridesharing [1,2,6] is more challenging as both passengers' queries and taxis' positions are highly dynamic and are real-time in most cases.

Besides, pricing mechanism is required to incite the driver. In Dial-a-Ride [7], vehicles need to return to the same location (depot) after the trip, which can be treated as the carpooling problem with additional restrictions about depot. In this paper, we adapt slugging as the ridesharing model to recommend whether and where should a user wait for ridesharing. Ridesharing recommendation is different from travelling plan problems [14] where the sequence of must-visit locations is known in advance and the target is to decide the optimal visit order. As an online recommendation problem, we need to provide ridesharing suggestions for each request in real time.

Other Issues. Though ridesharing is envisioned as a promising solution to mitigate traffic congestion and air pollution for metropolitan cities, people raise social discomfort and safety concerns about traveling with strangers. Social ridesharing with friends [14] is studied to overcome these barriers. Another interesting issue about ridesharing is fair payment mechanism, including when taking a ride with friends [15] and pricing mechanism to incite the taxi drivers in taxi ridesharing [1]. Trip grouping [16] is to group similar trips where the sources and destinations are close to each other according to certain heuristics. The difference is that there is no waiting time or cost constraints in trip grouping.

2.2 Recommender Systems

GPS records of taxis take down information including ID, time, longitude, latitude, speed, direction and occupation, which reflect the patterns of both passengers and taxi drivers. Applications based on taxi GPS trajectory data including urban planning [17,18], route prediction [19] and recommender systems [8–10,20]. We here focus on discussing recommender systems.

Current recommender systems provide services for either passengers or taxi drivers. A passenger-finding strategy based on L1-norm SVM [10] is proposed to determine whether a taxi should hunting or waiting for passengers. TaxiRec [8] evaluates the passenger-finding potentials of road clusters based on supervised learning and recommends the top-k road clusters for taxi drivers. T-Finder [9,20] recommends some locations instead road clusters by utilizing historical data for both passengers and taxis. There is no training process comparing to supervised learning techniques [8]. Comparing to recommending road clusters or locations for taxis or users [8,9,20], besides finding a taxi, ridesharing recommendation also needs to consider the possible destinations of the coming taxis to predict whether ridesharing can be successful or not. This makes the problem much more challenging.

3 Ridesharing Recommendation

3.1 Preliminaries

Road Network. We model a road network as a direct graph $G(V, E)$, where E and V are sets of road segments and intersections of road segments. The travel cost of each road segment (u, v) may be either time or distance measure. Since they can be converted from one to the other with the moving speed, they are used interchangeably. In addition, a grid index structure is built on the underlying road network. Given a location (lon, lat), we can find out a road segment on which the location is located.

Taxi. There are three possible status for a taxi: occupied $(\mathcal{O})$, cruising $(\mathcal{C})$ and parked $(\mathcal{P})$. A taxi can pick up a passenger $(\mathcal{P} \rightarrow \mathcal{O}$ and $\mathcal{C} \rightarrow \mathcal{O})$ or drop off a passenger $(\mathcal{O} \rightarrow \mathcal{P}$ and $\mathcal{O} \rightarrow \mathcal{C})$. Taxis mentioned in this paper refers to occupied taxis, i.e., taxis with passengers, and are willing to rideshare queries.

Ridesharing Query. A ridesharing query is defined as $Q = (id$, $timestamp$, l_s, l_d, t_s, t_e, $t_w)$ where id is the user id, $timestamp$ is when the query is submitted, l_s and l_d are respectively the source and the destination of the user, t_s is the maximum walking time from l_s to a new place; and t_e is the maximum walking time from the destination of taxi to l_d. t_w is the maximum waiting time for ridesharing at the new place for ridesharing.

Query Satisfaction. Q can rideshare with a taxi if and only if (i) the walking time to a new waiting place for the user is no longer than t_s; (ii) the walking time from the destination of the taxi to the user's own destination l_d is no longer than t_e; (iii) the waiting time at the new place for taxi is no longer than t_w.

Problem Definition. Given a ridesharing query $Q = (id$, $timestamp$, l_s, l_d, t_s, t_e, $t_w)$, we aim to recommend the user to rideshare or take a taxi directly by herself. For users that are recommended to rideshare with others, we also provide places that are easier to get ridesharing for users.

3.2 Road Segment Clustering

Since a single road segment is not a proper evaluation unit, we adapt the road segment clustering method proposed in [8], where k-means [11] is used to partition road segments[1] $\{r_1, r_2, ..., r_N\}$ into k clusters $\{C_1, C_2, ..., C_k\}$, to minimize the intra-cluster sum of square $\text{argmin} \sum_{i=1}^{k} \sum_{r_j \in C_i} ||r_j - \mu_i||$, where $\mu_i = \frac{1}{n_i} \sum_{r_j \in C_i} r_j$ and n_i is the number of road segments in the i-th cluster. Interesting readers may refer to the original paper [8] for the details of road segment clustering. Figure 3 shows the clustering result in the urban area of Nanjing when the number of clusters is set to 1,000. Since we built a grid index on the road networks, by identifying the road segment where location (lon, lat) located, we can get the cluster that the location belongs to.

3.3 Kernel Density Estimation

Kernel density estimation can be used with arbitrary distributions and does not assume a fixed distribution in advance, which is used to predict the probability for taxis to depart from somewhere in cluster C_i and have somewhere (lon, lat) as

[1] The mid-point of a road segment is considered as its representative point.

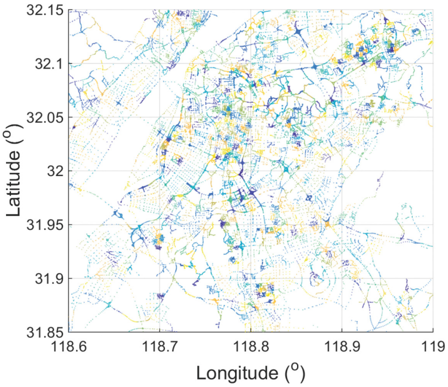


Fig. 3. Clustering on the road segments in urban area of Nanjing.

their destination. The estimation process consists of two steps: sample collection and distribution estimation.

Sample Collection. Recall that taxis departing from different clusters have different distributions of destinations since clusters have various features such as POI distributions. Given the GPS data, in order to personalize the destination distributions of each cluster, the start location decides which cluster will utilize the sample.

Intuitively ridesharing can succeed or not is related to not only locations of source and destination, but also when the query is submitted since traffic directions in modern city depend on time. Consider the traffics between companies and residencies, in the morning most traffics are likely to be from residencies to companies while in the evening traffics take the opposite direction. If only the destinations of GPS data are considered, we may recommend a user to rideshare even the query is from companies to residencies in the morning.

To avoid this, we collect samples as: $s = (lon, lat, TimeLabel)$ where (lon, lat) is the destination of the taxi and *TimeLabel* is the time label of the sample, indicating when the trip departing from a certain cluster to (lon, lat) happens. We divide a day into 48 time intervals, with the unit of 30 min (i.e., (1) 0:00 to 0:30, (2) 0:30 to 1:00, ..., (48) 23:30 to 0:00) and label them from 1 to 48. The label of the time interval containing the start time of the GPS data decides the *TimeLabel* of a sample. We discretize time because ridesharing consider taxis appearing in a time period to rideshare a query.

Distribution Estimation. Let $S^c =< s_1, s_2, ..., s_n >$ be the samples for a certain cluster c that follows an unknown density p. Its kernel density estimator over S^c for a new sample s_{n+1} is given by:

$$p(s_{n+1}|S^c) = \frac{1}{n\sigma^3}\sum_{i=1}^{n} K(\frac{s_{n+1} - s_i}{\sigma}), \tag{1}$$

where each sample $s_i = (lon_i, lat_i, TimeLabel_i)^T$ is a three-dimensional column vector with the longitude (lon_i) and latitude (lat_i) and $TimeLabel$ ($TimeLabel_i$). $K(.)$ is the kernel function, σ is the optimal bandwidth[2] [21]. In this paper we apply the widely used multi-dimensional normal kernel:

$$K(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})), \qquad (2)$$

where $\mathbf{x}$ is a real d-dimensional column vector and $\mathbf{x} {\sim} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We consider the time domain as the third dimension due to the nature of ridesharing.

3.4 Recommendation Framework

We now present the ridesharing framework in Algorithm 1. Given a ridesharing query $Q = (id, timestamp, l_s, l_d, t_s, t_e, t_w)$, network expansion technique [22] is used to select all reachable road segments from l_s in t_s time. Clusters that contain any of these road segments are added to the candidate cluster set $L = \{C_i | i = 1, ..., N\}$ (Line 1). Similarly we also get road segments that are reachable within t_e from l_d as $D = \{r_j | j = 1, 2, ..., M\}$ (Line 2). Denote $prob(C_i \rightarrow r_j)$ as the probability for taxis to depart from somewhere in cluster C_i and have somewhere on road r_j as their destination. For each candidate cluster C_i, we compute $prob(C_i \rightarrow r_j)$ for each road segment r_j in D with Eq. 1. The total probability P is used as the ridesharing potential of cluster C_i (Line 3 to 10). If no valid cluster exists for Q, the user will be suggested not to wait for ridesharing and take taxi directly (Line 12). The top-k clusters whose potentials are no less than a certain threshold are recommended for rideharing (Line 14).

4 Experiments

4.1 Experiment Settings

Dataset. The large-scale GPS data set is generated by over 7,000 taxis in a period of one month in Nanjing[3], China. Each GPS record includes ID, time, longitude, latitude, speed, direction and occupation. We only take the occupied trips into consideration to rideshare queries [1,2]. We divide data set into the training set and the testing set in terms of the start time of each record. In practice we can only utilize the past data to predict the future. We take data in June 1st – June 28th as training data. 1,000 records in June 29th are randomly selected as ridesharing queries. The start time, source and destination of trip are treated as *timestamp*, l_s and l_d in the queries.

Comparison of Performances. We compare our ridesharing recommendation with the naive "stay-as-where-you-are" strategy, denoted as RR and SAWYA

[2] Optimal bandwidth $\sigma = 0.969 n^{-\frac{1}{7}} \sqrt{\frac{1}{3}\sum_i s_{ii}}$ where s_{ii} is the marginal variance.

[3] Road networks are obtained from OpenStreetMap. http://www.openstreetmap.org.

Algorithm 1. Ridesharing recommendation

```
input  : $Q = (id, timestamp, l_c, l_d, t_s, t_e, t_w)$
output: Whether and where to wait for ridesharing
Get clusters $L = \{C_i | i = 1, ...., N\}$ that are reachable from $l_s$ in $t_s$ time;
Get roads $D = \{r_j | j = 1, ...., M\}$ that are reachable from $l_d$ in $t_e$ time;
foreach Candidate cluster $C_i$ in $L$ do
    $P = 0$;
    foreach Possible destination $r_j$ in $D$ do
        Take the mid-point ($r_j^{lon}$, $r_j^{lat}$) of $r_j$ and the label $TimeLabel$ of the
        time interval containing $timestamp$ as the new sample $s_{n+1} = (r_j^{lon}$,
        $r_j^{lat}$, $TimeLabel)$;
        Compute $prob(C_i \to r_j)$ with Eq. 1;
        $P+ = prob(C_i \to r_j)$;
    if $P \geq threshold$ then
        Add $C_i$ to answer set;
if Answer set is empty then
    Recommend user don't wait for ridesharing and take taxi directly;
else
    Sort according to $P$ and recommend top-$k$ clusters for users to rideshare.
```

respectively. In SAWYA, the users wait for ridesharing at where they are, i.e., the cluster that l_s is in. Recall in a ridesharing query we have walking time t_s and waiting time t_w, in SAWYA $t_s = 0$ and we define the new waiting time t'_w as $t_s + t_w$.

When evaluating the performance of SAWYA, during t'_w time, if there is any taxi whose source is in the same cluster as Q and destination is reachable from l_d of Q within t_e, SAWYA is considered to give users an accurate *to-rideshare*. Similarly, for each recommended cluster in our recommendation framework, if any taxi whose source is in them and destination is reachable from l_d of Q within t_e time, our recommendation is considered as an accurate *to-rideshare*. On the other hand, if a user is recommended not to wait for ridesharing and there is no taxi to rideshare, our recommendation is considered as an accurate *not-to-rideshare*.

Parameters. We study three parameters t_s, t_e and t_w about their influence on the performances of RR and SAWYA, as shown in Table 1. The walking speed is set to 1.4 m/s [23]. We recommend top-k clusters where k is set to 5.

4.2 Performance Metrics

Ridesharing Successful Ratio. We measure the ratio of successful ridesharing of both RR and SAWYA by *RSRatio*, defined as *RSRatio* = no. of accurate *to-rideshare* / no. of queries.

Table 1. Overview about parameters

Parameter	Default value	Range
t_s	4 min	[2 min, 4 min, 6 min, 8 min, 10 min]
t_e	4 min	[2 min, 4 min, 6 min, 8 min, 10 min]
t_w	4 min	[2 min, 4 min, 6 min, 8 min, 10 min]

Prediction Accuracy. We measure the accuracy of predicting whether the user should wait for ridesharing or not by *accuracy*, defined as *accuracy* = (no. of accurate *not-to-rideshare* + no. of accurate *to-rideshare*) / no. of queries.

Recommendation Accuracy. To evaluate the quality of cluster recommendations, it is important to find out how many clusters that actually have taxis to rideshare a query are discovered by our framework. For this purpose, we employ standard metrics, i.e., *precision* and *recall*:

$$precision = \frac{\text{no. of discovered clusters}}{\text{k}}, recall = \frac{\text{no. of discovered clusters}}{\text{no. of positive clusters}}.$$

Positive clusters are clusters with taxis to rideshare Q and discovered clusters are the positive clusters in the recommended clusters. Precision and recall are averages over all queries to obtain the overall performance.

4.3 Experiment Results

Effect of Walking Time t_s. Table 2 depicts the effect of walking time t_s from current locations of users to new places. Since users can walk to farther places, the number of candidate clusters increases for both RR and SAWYA, which relaxes the requirement of ridesharing. As the number of taxis to rideshare queries increases, both RR and SAWYA have better ridesharing successful ratio (*RSRatio*) and prediction accuracy (*Accuracy*). RR outperforms SAWYA in terms of *RSRatio* by at most 40 % since it can efficiently discover new places to rideshare for users. RR outperforms SAWYA in terms of *accuracy* by about 3 times since RR makes prediction for both *not-to-rideshare* and *to-rideshare*. This shows that about 500 users can save 4–8 min with RR. Given 9 RMB as the starting taxi fare and suppose users can save half of the total fare by ridesharing,

Table 2. Effect of walking time t_s (min).

Metrics	RR					SAWYA				
	2	4	6	8	10	2	4	6	8	10
RSRatio	0.143	0.196	0.234	0.297	0.310	0.122	0.141	0.173	0.203	0.215
Accuracy	0.695	0.710	0.726	0.738	0.751	0.122	0.141	0.173	0.203	0.215
Precision	0.316	0.317	0.356	0.395	0.433	-	-	-	-	-
Recall	0.875	0.867	0.746	0.615	0.516	-	-	-	-	-

Table 3. Effect of walking time t_e (min).

Metrics	RR					SAWYA				
	2	4	6	8	10	2	4	6	8	10
RSRatio	0.111	0.196	0.229	0.262	0.288	0.091	0.142	0.186	0.232	0.275
Accuracy	0.667	0.710	0.762	0.770	0.785	0.091	0.142	0.186	0.232	0.275
Precision	0.314	0.317	0.472	0.601	0.636	-	-	-	-	-
Recall	0.857	0.867	0.871	0.902	0.917	-	-	-	-	-

Table 4. Effect of waiting time t_w (min).

Metrics	RR					SAWYA				
	2	4	6	8	10	2	4	6	8	10
RSRatio	0.156	0.196	0.203	0.235	0.254	0.122	0.141	0.173	0.203	0.215
Accuracy	0.681	0.710	0.722	0.749	0.776	0.122	0.141	0.173	0.203	0.215
Precision	0.315	0.317	0.365	0.402	0.507	-	-	-	-	-
Recall	0.860	0.867	0.868	0.881	0.886	-	-	-	-	-

users can save 10.828-44.062 RMB by ridesharing. *Precision* increases as more positive clusters are discovered. *Recall* decreases since a longer t_s leads to more candidate clusters while we only recommend top-k to users.

Effect of Walking Time t_e. Table 3 depicts the effect of walking time t_e from the destinations of taxis to those of users. As t_e increases from 2 min to 10 min, more destinations of taxis become reachable for users, which increases the number of taxis to rideshare queries. Both RR and SAWYA achieve better ridesharing successful ratio (*RSRatio*) and prediction accuracy (*Accuracy*). *Precision* and *recall* both increase because the number of discovered clusters and positive clusters increase with more taxis to rideshare queries.

Effect of Waiting Time t_w. Table 4 depicts the effect of waiting time t_w at the waiting location. As t_w increases, users can wait for taxis for a longer time. Thus more taxis are taken into consideration and increase the probability to rideshare queries. The ridesharing successful ratio (*RSRatio*) and prediction accuracy (*Accuracy*) increase for both RR and SAWYA. As the number of taxis increases with t_w, both the number of discovered clusters and the number of positive clusters increases, leading to the increase in *precision* and *recall*.

5 Conclusion

In this paper, we proposed a recommendation framework based on kernel density estimation to predict whether and where should a user wait for ridesharing. In the framework, we grouped road segments into clusters and modeled the arrival patterns of taxis from different clusters. Given a query, we compute the potential of ridesharing by investigating the probabilities of possible destinations based on ridesharing requirements. Experimental results show that the accuracy about whether ridesharing or not and the ridesharing successful ratio are respectively about 3 times and at most 40 % better than the naive “stay-as-where-you-are” strategy. In the future work, we will study how to incorporate the influence of Point-of-Interests (POIs) into our ridesharing recommendation.

Acknowledgment. The author would like to thank Dr. Yanhua Li and Dr. Jia Zeng for providing their Nanjing taxi GPS dataset. The author also would like to thank anonymous reviewers and Dr. Cong Wang.

References

1. Ma, S., Zheng, Y., Wolfson, O.: Real-time city-scale taxi ridesharing. IEEE Trans. Knowl. Data Eng. **27**(7), 1782–1795 (2015)
2. Huang, Y., Bastani, F., Jin, R., Wang, X.S.: Large scale real-time ridesharing with service guarantee on road networks. PVLDB **7**(14), 2017–2028 (2014)
3. Ma, S., Wolfson, O.: Analysis and evaluation of the slugging form of ridesharing. In: SIGSPATIAL, pp. 64–73 (2013)
4. Kamar, E., Horvitz, E.: Collaboration and shared plans in the open world: studies of ridesharing. In: IJCAI, p. 187 (2009)
5. Calvo, R.W., de Luigi, F., Haastrup, P., Maniezzo, V.: A distributed geographic information system for the daily car pooling problem. Comput. OR **31**(13), 2263–2278 (2004)
6. Cao, B., Alarabi, L., Mokbel, M.F., Basalamah, A.: SHAREK: a scalable dynamic ride sharing system. In: MDM, pp. 4–13 (2015)
7. Attanasio, A., Cordeau, J., Ghiani, G., Laporte, G.: Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Comput. **30**(3), 377–387 (2004)
8. Wang, R., Chow, C.Y., Lyu, Y., Lee, V.C.S., Kwong, S., Li, Y., Zeng, J.: Taxirec: recommending road clusters to taxi drivers using ranking-based extreme learning machines. In: SIGSPATIAL (2015)
9. Yuan, N.J., Zheng, Y., Zhang, L., Xie, X.: T-finder: A recommender system for finding passengers and vacant taxis. IEEE Trans. Knowl. Data Eng. **25**(10), 2390–2403 (2013)
10. Li, B., Zhang, D., Sun, L., Chen, C., Li, S., Qi, G., Yang, Q.: Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. In: PerCom., pp. 63–68 (2011)
11. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
12. Cheng, C., Yang, H., King, I., Lyu, M.R.: Fused matrix factorization with geographical and social influence in location-based social networks. In: AAAI (2012)
13. Ye, M., Yin, P., Lee, W., Lee, D.L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: SIGIR, pp. 325–334 (2011)
14. Bistaffa, F., Farinelli, A., Ramchurn, S.D.: Sharing rides with friends: a coalition formation algorithm for ridesharing. In: AAAI, pp. 608–614 (2015)
15. Bistaffa, F., Filippo, A., Chalkiadakis, G., Ramchurn, S.D.: Recommending fair payments for large-scale social ridesharing. In: RecSys., pp. 139–146 (2015)

16. Gidófalvi, G., Pedersen, T.B., Risch, T., Zeitler, E.: Highly scalable trip grouping for large-scale collective transportation systems. In: EDBT, pp. 678–689 (2008)
17. Liu, Y., Kang, C., Gao, S., Xiao, Y., Tian, Y.: Understanding intra-urban trip patterns from taxi trajectory data. J. Geog. Syst. **14**(4), 463–483 (2012)
18. Zheng, Y., Liu, Y., Yuan, J., Xie, X.: Urban computing with taxicabs. In: UbiComp., pp. 89–98 (2011)
19. Yuan, J., Zheng, Y., Xie, X., Sun, G.: T-drive: Enhancing driving directions with taxi drivers' intelligence. IEEE Trans. Knowl. Data Eng. **25**(1), 220–232 (2013)
20. Yuan, J., Zheng, Y., Zhang, L., Xie, X., Sun, G.: Where to find my next passenger. In: UbiComp., pp. 109–118 (2011)
21. Silverman, B.W.: Density Estimation for Statistics and Data Analysis, vol. 26. CRC Press, Boca Raton (1986)
22. Papadias, D., Zhang, J., Mamoulis, N., Tao, Y.: Query processing in spatial network databases. In: VLDB, pp. 802–813 (2003)
23. Browning, R.C., Baker, E.A., Herron, J.A., Kram, R.: Effects of obesity and sex on the energetic cost and preferred speed of walking. J. Appl. Physiol. **100**(2), 390–398 (2006)