Route Recommendations for Intelligent Transportation Services

Yong Ge, Huayu Li and Alexander Tuzhilin

Abstract—The accumulated large amount of mobility data and the ability to track moving people or objects have enabled us to develop advanced mobile recommendations, which are essentially to recommend a sequence of locations to an individual user on the move. In this paper, we study a particular case of mobile recommendations, route recommendations to drivers, by utilizing vehicle GPS data. Specifically, we formulate a new Route Recommendation with Relaxed Assumptions (RR-RA) problem, the goal of which is to recommend a sequence of locations to a driver based on his current location in order to maximize his business success. To make our recommendation practical and scalable for real practice, we need to produce recommendation results in a timely fashion once a request emerges. Therefore, we propose an efficient algorithm to efficiently generate recommendations. Furthermore, we identify and address a destination-oriented route recommendation (DORR) problem. Without solving DORR problem, RR-RA alone does not work well in practice because drivers may encounter the destination constraint on a daily basis. We develop a dedicated and efficient algorithm for solving DORR problem. The package of solutions for both RR-RA and DORR problems provide a comprehensive approach for route recommendations to drivers. We evaluate our methods using both real-world GPS data and synthetic data, and demonstrate the effectiveness and efficiency of proposed methods with different evaluation metrics.

Index Terms—Recommender Systems, GPS Data, Mobile Recommendations, Intelligent Transportation

1 Introduction

Mobile recommendations aim to provide recommendations to users on the move, such as route recommendations to taxi drivers, POI recommendations to tourists. Mobile recommendations constitute a growing and important sub-area in the field of recommender systems [1] due to the following factors: (a) existence of a rich and diverse class of applications requiring mobile recommendations; (b) establishment and maturity of mobile infrastructure, including proliferation and wide adoption of various mobile devices that generate rich and useful data of high quality; (c) many mobile recommendation problems are still underexplored, and development of novel methods is required to solve these problems. All the aforementioned three factors constitute a perfect storm and make the emerging area of mobile recommendations an important and rich topic of research. Some of the manifestations of this phenomenon include a growing set of papers published on mobile recommender systems over the last few years [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] and the organization of workshops dedicated to this topic [12], [13].

In this paper, we focus on a particular aspect of this diverse and multifaceted field and study the problem of providing route recommendations to the drivers of different types of vehicles, including taxis, private cars, and tour buses. Such transportation application is important because it can: (a) recommend more efficient routes to drivers, especially the less experienced ones, and help them save time, money and possibly increase vehicle occupancy and utilization rates, (b) help economize fuel consumption which will result in energy savings and produce less traffic and pollution. Among the diverse class of transportation applications, the one that will stand to benefit the most from our study is taxi driving. This particular transportation application is important because taxis provide mobility in urban areas and play an important role in today's public transportation. However, current taxi service is still facing several practical challenges. First, although today's online taxi transportation companies (such as Uber, DiDi, Lyft) provide support to connect drivers with passengers, their solution mainly works as a request-based passive dispatch, where drivers are routed to different pick-up locations based on the real-time active pick-up requests by users. It does not pay much attention to potential passengers who may need services in near future. When there is no active pick-up requests nearby a empty cab, such request-based passive dispatch solution will not work, as it can not guide the empty cab to proactively search for future passengers. Second, most empty cab drivers plan their routes for searching for passengers based on their own experience when there is no service request available, which could lead to low occupancy rate. Effective and efficient recommendations of driving routes to drivers could increase this rate, which leads to more earnings and increased efficiency and effectiveness. To address these problems, Uber and Lyft are trying to develop some helpful solutions for drivers to better search for future passengers [14], [15], [16]. Their methods use physical sensing techniques to sense nearby potential passengers and send their location information to taxi cabs on which physical receiver are installed. Besides the privacy and cost issues [17], [18], these methods could not capture passengers who do not use their applications and cannot be detected.

To address these problems, we proposed and studied a mobile sequential recommendation (MSR) problem with fixed

• Yong Ge is with The University of Arizona.
E-mail: yongge@email.arizona.edu.
• Huayu Li is with Facebook. E-mail: lihuayu89@gmail.com.
• Alexander Tuzhilin is with New York University.
E-mail:atuzhili@stern.nyu.edu.

length constraint in [6]. Although our solutions provide a proactive way for taxi drivers to better search potential passenger(s), there are two important assumptions in that study. The first assumption is that we fix the number of pick-up locations of candidate route and only search optimal route from these ones. The second one is that the travel distance, when a driver cannot pick up any passenger after passing the last pick-up location of candidate route, is assumed to be a large constant and that this constant is the same for all candidate routes. However, both assumptions barely hold in real applications (we will provide more discussions about both assumptions in Section 3). Therefore, in this paper, we relax both assumptions and introduce a new route recommendation problem.

Specifically we formulate a new Route Recommendation with Relaxed Assumptions (RR-RA) problem, which essentially recommends a sequence of pick up locations to an empty cab based on potential pick up locations mined from historical taxi GPS data and current location of the empty cab. This recommendation is in the form of driving routes which makes the problem particularly challenging. We develop a novel approach to solve this problem. The goal of our approach is to generate driving routes that could minimize the travel distance before picking up passenger(s). While our intelligent route recommendation provides a new way for taxi drivers to improve their business performance, the computing process is very computationally expensive. It is challenging to make such recommendations practical for real taxi businesses because the results need to be produced online in a timely fashion to meet the real-time requests of drivers. Therefore, we develop a pruning-based algorithm for efficiently searching optimal route to recommend.

Furthermore, we identify and address an important destination constraint that taxi drivers may frequently face, and propose a destination-oriented route recommendation (DORR) problem in this paper. The DORR problem is crucial for any practically successful solution to driver's route recommendation problem. Without solving DORR problem, the solution for RR-RA can not be very practical for real application because drivers frequently run into DORR problem (we will explain this in detail in Section 4). Although DORR problem shares similar inputs as RR-RA problem, there is a destination constraint that has to be considered for route recommendation. Thus, we develop a new recommendation approach for solving DORR problem. As it also calls for efficient algorithms to make the online recommendation practical, we develop a dedicated and efficient algorithm to search for optimal route in this destination-constrained scenario. By solving DORR problem, we strengthen RR-RA problem and make our approaches more practical because the combined solutions for both RR-RA and DORR problems provide a comprehensive way for route recommendations to taxi drivers. We would like to note that as we focus on providing routing recommendations for empty-taxi drivers when there is no active pick-up request nearby, both RR-RA and DORR problems exclude the consideration of active pick-up requests.

We evaluate the efficiency of our methods for RR-RA and DORR problems using both real-world and synthetic data. The experimental results demonstrate that our methods could significantly save computing time for online recommendation compared with baseline methods. Furthermore, it is important to validate how effective the recommended routes are. While the ideal way for validating this is to conduct the field study (a.k.a., A/B testing) via collaborating with taxi and ride sharing companies, it is very difficult to provide such validations for several reasons. Therefore, we evaluate the effectiveness of our recommendations offline by using the traditional machine learning paradigm. The empirical results demonstrate that our route recommendations could greatly help drivers search for potential passengers, and reduce their driving without passenger by around 18%.

Overall, the contributions and implications of this paper are summarized as follows.

- We present a technically challenging and practically important RR-RA problem and develop an advanced method for solving this problem.
- We formulate a DORR problem and develop an advanced solution to solve it. By solving RR-RA and DORR problems, we are able to provide a "complete package" solution for route recommendations to taxi drivers.
- We conduct empirical tests to evaluate the performance of our methods with large-scale real-world taxi GPS data. We also use simulations to demonstrate practical value of our route recommendations.
- Our developed approaches shed some light on solving other transportation problems such as parking lot search and routing for carpool. The techniques developed for RR-RA and DORR problems will open a new range of methods in the recommendation field and the developed algorithms will lay a foundation for studying other mobile recommendation problems.

2 Related Work

2.1 Recommendations for Mobile Users

Recommender systems in the mobile environments have been studied before. The two surveys [1], [19] provide a broad overview of this prior work that can be classified into the research focused on mobile guides [19] and the other research that goes beyond the mobile guides [1].

Historically, the area of mobile recommendations started with the work on mobile guides that come in various "forms and shapes," and [19] provide an extensive categorization of the whole ranges of these guides according to their connectivity to the Internet, being indoor (e.g., museums) or outdoor, etc. Then [19] use this classification to develop a set of mobile guide design principles that can be used by application developers. The mobile guide systems typically use temporal, location-based and other types of contextual information (such as the current mood and interests of a tourist, weather information, etc.) obtained either from the user or extracted from the environment in order to query or search a certain repository of resources (such as, restaurants, museums and shows) and present the best matching resources (e.g., nearby restaurants that are currently open) to the user.

One of the early examples of research on mobile tourist guides is the Cyberguide project [20], which develops several tour guide prototypes for different hand-held platforms. [21] present a mobile tourist guide system COMPASS that supports many standard tourism-related functions, such as reserving a table at a restaurant or booking tickets for a show. COMPASS uses a similar type of the context-driven querying and searching approach [22] as the Cyberguide system [20]. Other examples of context-aware tourist guide systems proposed in the literature include the GUIDE system [23], the INTRIGUE system [24] and the MyMap system [25].

An alternative approach to mobile recommendations is based on the contextual preference elicitation and discovery [22], [26], [27], and it is being surveyed in [1]. As an example of this approach, the UbiquiTO system implementing a mobile tourist guide [28], provides intelligent adaptation of recommendations not only based on the specific location-based, temporal and other contextual information, but it also uses various rule-based and fuzzy set techniques to adapt the application content based on the user preferences and interests. Other examples of contextual preference elicitation and estimation approaches to mobile recommendations are also available in [29], [30], [31], [32]. Further, different combinations of these approaches could be achieved via a specially developed integration module that also features a graphical interface for mobile users [33]. Usefulness of such mobile recommendations is demonstrated in [34] where a study of 155,000 customers of a mobile portal reveal that the use of personalized recommendations instead of non-personalized ones leads to a significant increase in viewed and sold items in different navigational situations and to the overall sales increase.

2.2 Analytics of Taxi Trajectory Data

The data mining research literature has witnessed a growing interest in taxi trajectory data. The general goal of these works is to understand and discover behaviors and patterns that trajectories generate. For instance, Zhang et al. [35] analyze the taxi GPS data to study the impact of information on driver behavior and economic outcome, and find heterogeneity in individual learning behavior and driving decisions having economic impact. Liu et al. [36] study how a taxi driver gathers and learns information in an uncertain environment through the use of his social network with massive taxi GPS data. Although these studies reveal useful information and knowledge about taxi driver behaviors, they could not provide solution for navigating taxi drivers to earn more business. Le et al. [37] analyze the decision process from social perspective with trajectory data. Ge et al. [38] study the taxi driving fraud behavior (i.e., overcharging passengers by taking unnecessary detour) and develop detection methods for such fraud with taxi GPS data. Other examples of discovering useful pattern from trajectory data are also available in [39], [40], [41], [42].

2.3 Recommendations for Taxi Services

A few of research works have been done about recommendation for taxi services. Ge et al. [6] formulate a mobile sequential recommendation (MSR) problem, the objective of which is to recommend driving route for taxi drivers in order to minimize their driving distance without passenger(s). MSR shares the same goal as RR-RA problem that we study in this paper, but there are two important assumptions in MSR problem [6] as discussed in Section 1. To better solve MSR problem, Huang et al. [7] develop a new method for improving pruning and computational efficiency of search algorithm; Ye et al. [11] combine parallel computing and simulated annealing with novel global and local searches to further improve the computational efficiency of MSR problem. However, these methods in [11], [7], and [6] work based on the aforementioned two assumptions. Thus they cannot be used for solving our RR-RA or DORR problem. Yuan et al. [9] apply the constraint shortest path technique to find an optimal path between taxi driver's current position and one parking place. Ma et al. [8] propose a single-side and a dual side taxi searching technique for finding routes for additional passengers to share a taxi. Both techniques in [9] and [8] are designed to optimize a route between a starting point and an ending point, and thus they are not applicable to RR-RA problem. On the other hand, they did not address the direction constraint in our DORR problem, and thus they are not applicable to DORR problem either. In addition, Wang et al. [10] propose TaxiRec for route recommendation with road network information. Specifically, road network is first segmented into a number of road clusters, and then a ranking-based extreme learning machine (ELM) model is used to evaluate passenger-finding potential of each road cluster [10]. Since TaxiRec focuses on providing effective recommendation of top-k road segment clusters, rather than a sequence of pick-up locations, their solutions are not applicable to either RR-RA and DORR problem.

2.4 Vehicle Routing

This paper is also related to vehicle routing problem (VRP) that the operation research community has studies for decades [43]. The general goal of the VRP is to search optimal routes for vehicles to traverse in order to reach a set of customers. The traveling cannot be completed until all customers are reached. Our RR-RA and DORR problem are different from VRP in that it routes a driver for searching for next passenger, and it is not needed to reach all potential locations and the travel will stop once next passenger is reached. Due to such inherent differences between VRP and RR-RA/DORR problems, those solutions for the VRP could not be used for solving our RR-RA and DORR problems. Therefore, we develop novel data mining methods to solve RR-RA and DORR problems. In addition, while the VRP is often resolved offline [43], RR-RA and DORR problems need to be solved online by using the real-time location information of drivers. The efficiency of solution for RR-RA and DORR problems is much more important than that for VRP problem. Therefore, in this paper we aim to develop novel and efficient algorithms for solving both problems.

3 Route Recommendation with Relaxed Assumptions

In this section, we introduce a new problem, i.e., Route Recommendation with Relaxed Assumptions (RR-RA). Unlike the problem we studied in [6], we relax two important assumptions to make the problem more practical for real applications.

3.1 Problem Statement

Given an empty cab's location $PoCab$, there are a set of N potential pick-up locations denoted as $\mathcal{C} = \{C_1, C_2, \cdots, C_N\}$ nearby $PoCab$. For each pick-up location, there is estimated pick-up probability, denoted as P_i. Let us use $\mathcal{P}$ to denote the set of probabilities $\mathcal{P} = \{P_i\}(i = 1, \cdots, N)$, where $P_i(i = 1, \cdots, N)$ is assumed to be independent to each other. The objective of RR-RA problem is to search the optimal route that has minimum expected driving distance (EDD) before picking up passenger(s) and includes less than L pick-up locations. Each candidate route is essentially a sequence of potential pick-up locations. Following the same practice in [6], we assume that all potential pick-up locations in each candidate route are different. Let us use $\overrightarrow{R}$ to denote one candidate route, and use $D(\overrightarrow{R})$ to represent the expected driving distance before pick-up event while a driver taking candidate route $\overrightarrow{R}$. Based on these notation, RR-RA problem is formally stated as follows.

RR-RA Problem
Given: An empty cab's location $PoCab$, a set of N potential pick-up points denoted as $\mathcal{C} = \{C_i\}(i = 1, \cdots, N)$ nearby $PoCab$, and a set of pick-up probability at these N locations denoted as $\mathcal{P} = \{P_i\}(i = 1, \cdots, N)$.
Objective: Recommending an optimal route $\overrightarrow{\mathbb{R}}$ for the empty cab, which minimizes expected driving distance $D(\overrightarrow{R})$ before pick-up event and includes less than L pick-up locations ($L \leq N$).

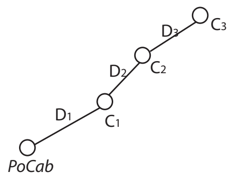
Compared with MSR problem studied in [6] that only searches optimal route with fixed L pick-up locations, we relax this constraint in RR-RA problem. Such a relaxed assumption brings two consequences. First, we make the recommendation problem more practical because taxi drivers may want to explore driving routes with different number of pick-up locations, rather than a fixed number of ones. The recommendation results will better meet the practical needs of taxi drivers than those of MSR problem. Second, the search space of RR-RA problem is much larger than that of MSR in [6], which leads to more computational complexity. Furthermore, we relax another assumption that we made in the definition of PTD function in [6]. We will introduce this relaxation of assumption in Section 3.3. Due to relaxing both assumptions, the algorithms we developed in [6] actually cannot work for RR-RA problem. Thus, we will develop a dedicated method to solve RR-RA problem in this paper.

There are two challenges to solve RR-RA problem. The first one is how to obtain potential pick-up locations and pick-up probability given an empty cab requesting recommendation service. Second, it is how to timely search the optimal route from all potential candidates to meet the real-time request of route recommendation. In fact, if the computational cost for one candidate route is $Cox(\overrightarrow{R}) = 1$, the computational complexity of RR-RC problem is $\mathcal{O}(N!)$. In the following two subsections, we introduce our solutions for both challenges.

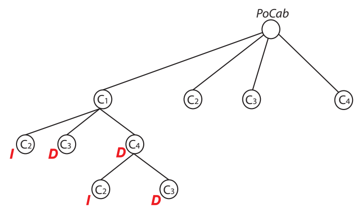
3.2 Generating Pick-up Location and Probability

In this section, we introduce how to generate potential pick-up locations (i.e., $\mathcal{C}$) and estimate pick-up probability at each one (i.e., $\mathcal{P}$) when the time and location of recommendation request by one empty cab is given.

Generating Pick-up Location. Given the location $PoCab$ and time t (e.g., $2PM$ on Monday) of one empty cab that requests the recommendation of driving route, we will extract



(a) A Sample Route



(b) A Sample Tree Structure

Fig. 1: Two Illustration Examples

a set of raw pick-up points from historical taxi GPS data[1]. Specifically, as long as one pick-up event happens within the time range $[t - \alpha, t + \alpha]$ on the same week day (e.g., Monday) and the corresponding pick-up point locates within a spatial range that has center as $PoCab$ and radius as τ_1, we will extract this pick-up point. After getting all these raw pick-up points, we will group them into different clusters based on the driving distance calculated via Google Map APIs. Finally, the centroids of these clusters are considered as potential pick-up locations that will be used for generating recommendations.

Estimating Pick-up Probability. For each candidate pick-up location (i.e., the centroid of pick-up location cluster), we will estimate the probability of pick-up event around time t (e.g., $2PM$ on Monday) based on historical data. The idea is to measure how frequent pick-up event happens when cabs travel across a pick-up location cluster C_i around time t on the same week day. Specifically, we first count the number of empty cabs that enter the spatial coverage of cluster C_i within the time range $[t - \alpha, t + \alpha]$ on the same week day (e.g., Monday) from historical data, and denote it as $\#_T$. Among these $\#_T$ empty cabs, we count the number of pick-up events $\#_P$ that happen within the spatial coverage of cluster C_i among the same time range. Finally, the probability of pick-up event at C_i around time t is estimated as $P_i^t = \frac{\#P}{\#T}$.

3.3 Method for Computing Optimal Route

First let us introduce how we define $D(\overrightarrow{R})$ for a candidate route $\overrightarrow{R}$. To simplify the introduction, let us consider an example of candidate route $\overrightarrow{R} = PoCab \to C_1 \to C_2 \to C_3$ as shown in Figure 1(a). When a driver takes this route, he will have probability P_i $(1 \leq i \leq 3)$ of picking up passenger(s) at each location C_i $(1 \leq i \leq 3)$. The driving distance before picking up passenger could be D_1 with probability P_1, $D_1 + D_2$ with probability $\bar{P}_1 P_2$, or $D_1 + D_2 + D_3$ with probability $\bar{P}_1 \bar{P}_2 P_3$, where $\bar{P}_i$ is defined as $\bar{P}_i = 1 - P_i$. In addition, the probability of picking up no passenger by taking this route is $\bar{P}_1 \bar{P}_2 \bar{P}_3$. We denote the driving distance beyond the last pick-up location of candidate route $\overrightarrow{R}$ as $\tilde{D}_3$. In the definition of PTD function in [6], we assume $\tilde{D}_3$ is a very big constant and it is the same at the last pick-up location of any candidate route. Based on this assumption, we identify the monotone property of PTD function in [6], which leads us to develop efficient algorithms for solving MSR problem.

1. Taxi GPS data records its location, timestamp and status every seconds. The status is either occupied or empty. When the status changes from empty to occupied, the corresponding location is a raw pick-up point

However, this assumption barely holds in reality. First, as $\tilde{D}_i$ denotes the distance that a taxi driver needs to drive to find passenger from location C_i, it could be different at different pick-up location C_i. Second, the value of $\tilde{D}_i$ may not be big at some locations such as those around crowd POIs (e.g., station, hotel). Therefore, we relax this assumption in this paper to make the problem more practical. Specifically, we leverage historical taxi GPS data to estimate $\tilde{D}_i$ for each pick-up location C_i offline. Instead of being a large constant, the estimated value of $\tilde{D}_i$ ($1 \leq i \leq N$) may vary at different locations and around different times. Given this relaxation, the monotone property of PTD function in [6] will not hold, and thus pruning algorithms developed in [6] will not work. Given the above statement, we derive the expected driving distance (EDD) $D(\overrightarrow{R})$ before pick-up event while a driver taking route $\overrightarrow{R} = PoCab \to C_1 \to C_2 \to C_3$ as:

$$D(\overrightarrow{R}) = D_1P_1 + (D_1 + D_2)\bar{P}_1P_2 + (D_1 + D_2 + D_3)\bar{P}_1\bar{P}_2P_3 + \tilde{D}_3\bar{P}_1\bar{P}_2\bar{P}_3. \tag{1}$$

To search optimal route for a taxi driver, a straightforward method needs to compute $D(\overrightarrow{R})$ for each possible candidate route with less than L pick-up locations, and then pick the best one $\mathbb{R}$ that has minimum $D(\overrightarrow{R})$. This will involve a lot of computation. In the following, we introduce important monotone property of $D(\overrightarrow{R})$, which allows us to prune many candidate routes without computing $D(\overrightarrow{R})$. Such pruning will be done offline before the real-time request of route recommendation occurs, and thus it will save us a lot of time for real-time recommendation.

To introduce the monotone property, let us consider two general candidate routes $\overrightarrow{R^1} = PoCab \to C_1 \cdots \to C_i$ and $\overrightarrow{R^2} = PoCab \to C_1 \cdots \to C_i \to C_{i+1}$. The only difference of $\overrightarrow{R^2}$ than $\overrightarrow{R^1}$ is that one more pick-up location C_{i+1} is appended. $D(\overrightarrow{R^1})$ and $D(\overrightarrow{R^2})$ can be derived as follows.

$$D(\overrightarrow{R^1}) = D_1P_1 + (D_1 + D_2)\bar{P}_1P_2 + \cdots + \sum_{ii=1}^{i} D_{ii}P_i \prod_{ii=1}^{i-1} \bar{P}_{ii} + \tilde{D}_i \prod_{ii=1}^{i} \bar{P}_{ii}. \tag{2}$$

$$D(\overrightarrow{R^2}) = D_1P_1 + (D_1 + D_2)\bar{P}_1P_2 + \cdots + \sum_{ii=1}^{i} D_{ii}P_i \prod_{ii=1}^{i-1} \bar{P}_{ii} + \sum_{ii=1}^{i+1} D_{ii}P_{i+1} \prod_{ii=1}^{i} \bar{P}_{ii} + \tilde{D}_{i+1} \prod_{ii=1}^{i+1} \bar{P}_{ii}. \tag{3}$$

We then derive the difference between $D(\overrightarrow{R^2})$ and $D(\overrightarrow{R^1})$ as follows.

$$\begin{aligned} \Delta D &= D(\overrightarrow{R^2}) - D(\overrightarrow{R^1}) \\ &= \prod_{ii=1}^{i} \bar{P}_{ii}\{P_{i+1}\sum_{ii=1}^{i+1} D_{ii} + \bar{P}_{i+1}\tilde{D}_{i+1} - \tilde{D}_i\} \\ &= \prod_{ii=1}^{i} \bar{P}_{ii}\{P_{i+1}(\sum_{ii=1}^{i+1} D_{ii} - \tilde{D}_{i+1}) + (\tilde{D}_{i+1} - \tilde{D}_i)\} \end{aligned} \tag{4}$$

From Equation 4, we could conclude the following two monotone property: (a) if $\sum_{ii=1}^{i+1} D_{ii} > \tilde{D}_{i+1}$ and $\tilde{D}_{i+1} > \tilde{D}_i$, ΔD should be bigger than 0, i.e., $D(\overrightarrow{R^2}) > D(\overrightarrow{R^1})$; (b) if $\sum_{ii=1}^{i+1} D_{ii} < \tilde{D}_{i+1}$ and $\tilde{D}_{i+1} < \tilde{D}_i$, ΔD should be less than 0, i.e., $D(\overrightarrow{R^2}) < D(\overrightarrow{R^1})$. For the convenience of presentation, let us denote this two property as $PR(a)$ and $PR(b)$. In other words, when we append one more pick-up location to a candidate route, we may conclude EDD value will increase if the condition of $PR(a)$ is satisfied, or will decrease if the condition of $PR(b)$ is satisfied.

Next we introduce how we use these two property to prune candidate routes in advance without computing EDD function. To make the introduction easy to follow, let us consider a set of 4 pick-up locations and specify $L = 3$. There are totally 24 possible candidate routes with less than 3 pick-up locations. In Figure 1(b), we show partial possible candidate routes with a tree structure. Let us consider route $PoCab \to C_1$. As there is only one pick-up location, there is no need to apply property $PR(a)$ and $PR(b)$. Next let us consider all routes that include one more appended pick-up location to route $PoCab \to C_1$, which include $PoCab \to C_1 \to C_2$, $PoCab \to C_1 \to C_3$, and $PoCab \to C_1 \to C_4$. We could check the conditions of property $PR(a)$ and $PR(b)$ for each of these three. Take route $PoCab \to C_1 \to C_2$ as an example, if $D_1 + D_2 > \tilde{D}_2$ and $\tilde{D}_2 > \tilde{D}_1$, we can conclude EDD of route $PoCab \to C_1 \to C_2$ will increase compared with that of route $PoCab \to C_1$. Let us use I to tag this conclusion at pick-up location C_2. This tag indicates that route $PoCab \to C_1 \to C_2$ can not be optimal route because route $PoCab \to C_1$ is better than it, thus we do not need to compute EDD for route $PoCab \to C_1 \to C_2$. Similarly, if the condition of $PR(b)$ (i.e., $D_1 + D_2 < \tilde{D}_2$ and $\tilde{D}_2 < \tilde{D}_1$) is satisfied, we use D to tag the conclusion at C_2, i.e., EDD of route $PoCab \to C_1 \to C_2$ will decrease compared with that of route $PoCab \to C_1$. The tag D indicates that route $PoCab \to C_1$ can not be optimal because route $PoCab \to C_1 \to C_2$ is better than it, thus we do not need to compute EDD for route $PoCab \to C_1$.

Given $PoCab$, N potential pick-up locations and L, we will first generate all possible candidate routes and store them with a tree structure as shown in Figure 1(b). We will then conduct the same process over all candidate routes (from ones with 1 pick-up locations to those with L ones) on the tree structure. Finally we will be able to tag partial pick-up locations with either I or D. Please note that many pick-up locations may not have any tag if conditions of both $PR(a)$ and $PR(b)$ are neither satisfied, and that we do not need to tag any pick-up location which appears as the first one of candidate route. In Figure 1(b), we show some hypothetic tags for partial pick-up locations of shown candidate routes. Based on these tags, we can conduct the following two pruning strategies: (a) if I is tagged to one pick-up location C_i, the candidate route ending with C_i can be pruned, i.e., removing from consideration for optimal route; (b) if D is tagged to one pick-up location C_i, the candidate route ending with the parent node (i.e., pick-up location) of C_i can be pruned. As many candidate routes may be pruned based both strategies and pruning takes less time, we will be able to save a lot of computation of EDD for them.

In fact, we can build the tree structure as shown in Figure 1(b) offline before we know the position of an empty cab requesting recommendation service. We could obtain N

potential pick-up locations at different times in a region offline based on historical data. We then build the tree structure (excluding $PoCab$) with these locations and user-specified L. Based on the tree structure, we could conduct the same pruning strategies as described above. Once $PoCab$ of one empty cab is given, we can search the optimal one from remaining candidate routes. Since many candidate routes have been pruned offline, we will only need to compute EDD for remaining candidate routes and pick the optimal one with minimum EDD. Thus, we will save a lot of computation for the online recommendation.

4 Destination-Oriented Route Recommendations (DORR)

In this section, we introduce a destination-oriented route recommendation (DORR) problem. Compared to RR-RA problem, there is a destination-driven direction constraint in DORR problem, which makes the methods for RR-RA problem not work.

4.1 Problem Statement

In the real-world taxi business, a driver of empty cab may want to quickly go to an area from a far-away location and he may also expect opportunity to pick up a customer on the way back. A taxi driver may have such desire under several scenarios. For instance, a driver in New York City (NYC) may usually operate in the downtown area; after delivering a passenger to a hotel in the uptown area, he wants to quickly drive back to the downtown area and expects to earn some business on the way back. In fact, in big urban areas, many taxi drivers usually have their typical or familiar operation areas and they tend to drive back after they happen to travel to a far-away location. As another scenario, many taxi drivers in metropolitan areas live in places which are often far away from their operation areas, and they want to pick up customers on the way to their operation area when they leave home and start their work daily. Since a particular driver usually has a destination constraint during his driving, we name this practical challenge as a destination-orient route recommendation (DORR) problem.

We observe two main constrains imbedded in DORR problem. First, we have to take into account the specific destination where a driver wants to go back for addressing this DORR problem. Second, in the destination-constrained scenario, drivers have more concern about the driving direction which may be detoured after picking up a customer, because they want to go back to their operation areas within some reasonable time. The request-based passive dispatch solution provided by today's online taxi transportation companies (e.g., Uber and DiDi) could not work in this scenario because they have not considered both constraints at all. To meet this specific need, we develop a dedicated method for addressing this DORR problem. In the following, let us first state this DORR problem in a formal way.

Let us denote one location as $\mathcal{E}$ which represents an area where the driver D_i usually operates, and another location as $\mathcal{S}$ which is a place that is far away from $\mathcal{E}$, and the driver D_i wants to go back to $\mathcal{E}$ from $\mathcal{S}$. From historical data, we could obtain a set of N potential pick-up points, $\mathbb{C} = \{C_1, C_2, \cdots, C_N\}$, along the way from $\mathcal{S}$ to $\mathcal{E}$, and the estimated pick-up probability P_i associated with each pick-up point C_i. Let $\mathbb{P} = \{P_1, P_2, \cdots, P_N\}$ denote the probability set. In additional to the probability, we could also get the direction information associated with each pick-up point. For each pick-up point C_i, we first gather all historical pick-up events T^{C_i} at this pick-up point. We denote all pick-up events associated with the pick-up point set $\mathbb{C}$ as $\mathbb{T}^{\mathbb{C}}$. Each pick-up event is actually a location trace, which indicates that a taxi driver delivers passenger(s) from this pick-up point to another location. We represent each pick-up event as a directed arrow staring from this pick-up point as shown in Figure 2 (a). This arrow reflects the direction toward a particular location where the driver drops off passenger(s). For each pick-up point C_i, we summarize the potential direction of pick-up events happening at C_i into 8 bins as shown in Figure 2 (b).

Comparing with the direct travel from $\mathcal{S}$ to $\mathcal{E}$, it will certainly increase the travel distance for driver D_i to intentionally pass several pick-up points on the way from $\mathcal{S}$ to $\mathcal{E}$. We can calculate this certainly increased travel distance based on the passed pick-up points. We denote the certainly increased travel distance as certain cost. Furthermore, when driver D_i picks up passenger(s) at a pick-up point on his way from $\mathcal{S}$ to $\mathcal{E}$, this pick-up event may cause the driver to travel more distance. We denote such potentially increased travel distance as uncertain cost. Given all pick-up events associated with one pick-up point, we are able to estimate this uncertain cost. We first compute the exactly-increased travel distance for each historical pick-up event, such as a pick-up event at A in Figure 3 (a). Then the average of exactly-increased travel distance over all pick-up events at one pick-up point may be an estimation of such uncertain cost. Thus, there are such two aspects of cost associated with each pick-up point. Of course, the driver has some chance to earn business by passing pick-up points. Given all pick-up points and their associated information, we can generate many candidate routes consisting of different pick-up points from $\mathcal{S}$ to $\mathcal{E}$. For example, in Figure 3 (a), we can have sequences like $\mathcal{S} \rightarrow A \rightarrow B \rightarrow E$ or $\mathcal{S} \rightarrow A \rightarrow C \rightarrow D \rightarrow E$. We represent the set of all possible routes as $\mathbb{R}$, each of which leads to different uncertain and certain cost, and business success.

Our goal is to search an optimal sequence of pick-up points for a taxi driver in order to maximize his business success with constraint on certain and uncertain cost. Let us use a function $\mathcal{G}$ to assess the two aspects of cost for each candidate route. For example, for a particular sequence R_i ($R_i \in \mathbb{R}$), we denote the function as $\mathcal{G}(R_i, S, E, P_{R_i}, T^{R_i})$, where T^{R_i} and P_{R_i} represent the associated pick-up events and probability information of R_i. In other words, for a specific sequence, the function $\mathcal{G}$ depends on source and end nodes, pick-up points along the sequence, and all associated information of these pick-up points. The business success of taxi driver in this destination-constrained scenario may be measured in different ways. In Section 4.2, we will introduce two ways for measuring the business success and the specific formula of function $\mathcal{G}$. Given the above statement, we can formally define the DORR problem as follows.

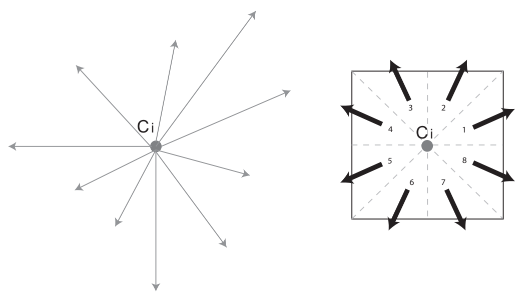


Fig. 2: Pick-up Events and Direction Summarization

The DORR Problem
Given: An original location $\mathcal{S}$ and an destination location $\mathcal{E}$, a pick-up point set $\mathbb{C}$ and associated probability set $\mathbb{P}$, the associated pick-up event set $\mathbb{T}_{\mathbb{C}}$, a taxi driver who locates at $\mathcal{S}$ and wants to go back to $\mathcal{E}$ quickly, a cost constraint TC that is essentially to limit the potentially travel distance.
Objective: Recommending an optimal sequence of pick-up points for the taxi driver, which could maximize the business success under the given cost constraint.

Similar as RR-RA problem, we assume that a driver would not visit a pick up location more than once in the DORR problem. In other words, there is no duplicate pick-up location in each candidate route.

To solve this problem, we have the same two challenges as those of RR-RA problem: (1) how to obtain reliable pick-up locations and associated probability, and (2) how to efficiently search optimal driving routes. We will use almost the same method described in Section 3.2 to solve the first challenge. The only difference is we will set a different spatial range based on the two locations, $\mathcal{S}$ and $\mathcal{E}$, to extract raw pick-up locations. Specifically, let us denote the latitude and longitude of $\mathcal{S}$ and $\mathcal{E}$ as $(lat_{\mathcal{S}}, lng_{\mathcal{S}})$ and $(lat_{\mathcal{E}}, lng_{\mathcal{E}})$, respectively. We consider the raw pick-up points within the spatial range that has the centre as $(\frac{lat_{\mathcal{S}}+lat_{\mathcal{E}}}{2}, \frac{lng_{\mathcal{S}}+lng_{\mathcal{E}}}{2})$ and the radius as τ_2. Suppose the computational cost for one candidate route is $Cox(\mathcal{G}) = 1$. The computational complexity of this DORR problem is $\mathcal{O}(N!)$. In the following section, we introduce the proposed method for solving the second challenge.

4.2 Cost Function and Business Success

In this section, we introduce the formulation of function $\mathcal{G}$ and the measurements of business success.

Suppose there are totally K historical pick up events happening at C_i. Let us denote each of them as $T_k^{C_i}$ $(1 \le k \le K)$. Also let us denote the corresponding drop off location of each pick up event as $Dr_k^{C_i}$ $(1 \le k \le K)$. If a driver only visits one pick up location C_i while traveling from $\mathcal{S}$ to $\mathcal{E}$, he may pick up a customer at C_i and deliver this passenger to different locations. Our idea is to leverage the average of historical K pick-up events to estimate the travel distance. Thus driver's driving route could be $\mathcal{S} \to C_i \to Dr_k^{C_i} \to E$ $(1 \le k \le K)$. We represent the driving distance of each route as $D_k(C_i)$ $(1 \le k \le K)$. The average of K routes is $D(C_i) = \sum_{k=1}^{K} D_k(C_i)/K$, which is used to estimate the driver's potential driving distance if he picks up a customer at C_i. In fact, as the travel distance $D_{\mathcal{S}\to C_i}$ from $\mathcal{S}$ to C_i remains the same for each pick up event $T_k^{C_i}$ $(1 \le k \le K)$ at C_i, we may rewrite $D(C_i)$ as $D(C_i) = D_{\mathcal{S}\to C_i} + \overline{D}^u_{C_i\to\mathcal{E}}$, where $\overline{D}^u_{C_i\to\mathcal{E}}$ denotes the average driving distance from C_i to $\mathcal{E}$ when the driver picks up a customer at C_i. It can be estimated with the historical pick up events at C_i as well. Let us use $D_{C_i\to Dr_k^{C_i}\to E}$ $(1 \le k \le K)$ denote the distance of each pick event k. Then $\overline{D}^u_{C_i\to\mathcal{E}}$ could be computed as $\overline{D}^u_{C_i\to\mathcal{E}} = \sum_{k=1}^{K} D_{C_i\to Dr_k^{C_i}\to E}/K$.

To simplify the further introduction about cost function $\mathcal{G}$, let us look at an example of driving route: $\mathcal{S} \to C_1 \to C_2 \to C_3 \to \mathcal{E}$. When a driver takes this sample route, he may pick up a customer at C_1, C_2 and C_3 with probability as P_1, $(1-P_1)P_2$, and $(1-P_1)(1-P_2)P_3$, respectively. We get the cost function $\mathcal{G}$ for estimating the potential driving distance as: $\mathcal{G} = P_1 D(C_1) + (1-P_1)P_2 D(C_2) + (1-P_1)(1-P_2)P_3 D(C_3) + (1-P_1)(1-P_2)(1-P_3)D$, where D represents the driving distance if a driver does not pick up a customer at any of three locations. In other words, D is the actual driving distance along the route $\mathcal{S} \to C_1 \to C_2 \to C_3 \to \mathcal{E}$. $D(C_2)$ and $D(C_3)$ are the estimated driving distances when the driver picks up a customer at C_2 and C_3, respectively.

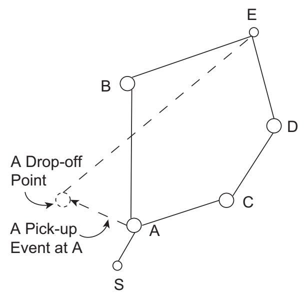
When a driver follows a route $\mathcal{S} \to C_1 \to C_2 \to C_3 \to \mathcal{E}$, the probability of picking up passenger(s) on the route is $1-(1-P_1)(1-P_2)(1-P_3)$, which is the measurement of business success used in this paper. Alternatively, the potential earning could also be one measurement of business success. For instance, if there are totally K historical pick up events at C_i, let us denote the earning of each historical pick up event as $E_k^{C_i}$. The average of earnings at C_i is $E(C_i) = \sum_{k=1}^{K} E_k^{C_i}/K$. Consequently, for the example route $\mathcal{S} \to C_1 \to C_2 \to C_3 \to \mathcal{E}$, the potential earning can be estimated as $P_1 E(C_1) + (1-P_1)P_2 E(C_2) + (1-P_1)(1-P_2)P_3 E(C_3)$.

4.3 Computing Algorithm

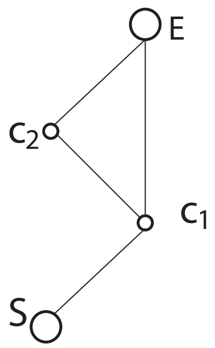
In this section, we present an efficient computing algorithm for searching the optimal route of DORR problem.

If a driver takes a route $\mathcal{S} \to C_1 \to \mathcal{E}$ as shown in Figure 3 (b), the potential driving distance D_1 for this route can be denoted as $D_1 = P_1(D_{\mathcal{S}\to C_1} + \overline{D}^u_{C_1\to\mathcal{E}}) + (1-P_1)(D_{\mathcal{S}\to C_1} + D_{C_1\to\mathcal{E}})$ based on the defined function $\mathcal{G}$, where $D_{\mathcal{S}\to C_1}$ (or $D_{C_1\to\mathcal{E}}$) is driving distance from $\mathcal{S}$ to C_1 (or from C_1 to $\mathcal{E}$). If the driver takes another route $\mathcal{S} \to C_1 \to C_2 \to \mathcal{E}$, which is obtained by simply adding a pick up location after C_1, the potential driving distance D_{12} for this route can be similarly represented as $D_{12} = P_1(D_{\mathcal{S}\to C_1} + \overline{D}^u_{C_1\to\mathcal{E}}) + (1-P_1)P_2(D_{\mathcal{S}\to C_1} + D_{C_1\to C_2} + \overline{D}^u_{C_2\to\mathcal{E}}) + (1-P_1)(1-P_2)(D_{\mathcal{S}\to C_1} + D_{C_1\to C_2} + D_{C_2\to\mathcal{E}})$. By carefully deriving, we may find that D_{12} can not be smaller than D_1. In fact, we may obtain the following general monotone property.

Lemma 1. Given a candidate route $R_1 = \mathcal{S} \to C_1 \to \cdots \to C_i \to \mathcal{E}$, and another candidate route with an added pick up location following C_i as $R_2 = \mathcal{S} \to C_1 \to \cdots \to C_i \to C_{i+1} \to \mathcal{E}$, the potential driving distance for taking R_1 and R_2 is denoted as $D_{1,\cdots,i}$ and $D_{1,\cdots,i+1}$ respectively. Then we can infer $D_{1,\cdots,i} \le D_{1,\cdots,i+1}$. In other words, the potential driving distance can not decrease when one more pick up location is appended to a candidate route. (We provide detailed proof for this lemma in online supplement.)



(a) An Illustration Example



(b) Route Samples

Fig. 3: Illustration Examples for DORR

This monotone property could greatly help us save the computational cost when we search the optimal sequence of pick up locations for a taxi driver in the real time, because after we conclude that the potential driving distance of one candidate route R is over the threshold of cost (i.e.,TC), we do not need to consider those extended candidate routes which include the candidate route R in the beginning.

Furthermore, we identify another monotone property that is also useful for computing potential driving distance. Let us still take the route $\mathcal{S} \rightarrow C_1 \rightarrow C_2 \rightarrow \mathcal{E}$ in Figure 3 (b) as an example, where we have $D_{12} = P_1(D_{\mathcal{S}\rightarrow C_1} + \overline{D}^u_{C_1\rightarrow\mathcal{E}}) + (1-P_1)P_2(D_{\mathcal{S}\rightarrow C_1} + D_{C_1\rightarrow C_2} + \overline{D}^u_{C_2\rightarrow\mathcal{E}}) + (1-P_1)(1-P_2)(D_{\mathcal{S}\rightarrow C_1} + D_{C_1\rightarrow C_2} + D_{C_2\rightarrow\mathcal{E}})$. We identify that D_{12} will monotonely increase as $D_{C_1\rightarrow C_2}$, $\overline{D}^u_{C_2\rightarrow\mathcal{E}}$ or P_2 increases. In general, we have the following monotone property.

Lemma 2. Given a candidate route $R = \mathcal{S} \rightarrow C_1 \rightarrow \cdots \rightarrow C_{i-1} \rightarrow C_i \rightarrow \mathcal{E}$, the potential driving distance for taking route R is denoted as $D_{1,\cdots,i}$. We can infer that $D_{1,\cdots,i}$ monotonely increases as $D_{C_{i-1}\rightarrow C_i}$, $\overline{D}^u_{C_i\rightarrow\mathcal{E}}$ or P_i increases. (We provide detailed proof for this lemma in online supplement.)

With this monotone property, we may be able to prune one candidate route by first checking these three values associated with the last pick up location. For instance, given two candidate routes $R_1 = \mathcal{S} \rightarrow C_1 \rightarrow\rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow \mathcal{E}$ and $R_2 = \mathcal{S} \rightarrow C_1 \rightarrow\rightarrow C_2 \rightarrow C_3 \rightarrow C_5 \rightarrow \mathcal{E}$, we can derive that the potential driving distance of R_2 is equal to or bigger than that of R_1 if we have $D_{C_3\rightarrow C_4} \leq D_{C_3\rightarrow C_5}$, $\overline{D}^u_{C_4\rightarrow E} \leq \overline{D}^u_{C_5\rightarrow E}$ and $P_4 \leq P_5$. Thus we do not need to compute function $\mathcal{G}$ for R_2 if we already concluded the potential driving distance of R_1 is over the threshold of cost.

Based on the two monotone properties, we will first enumerate and check candidate routes with 1 pick up location. Then we extend previously-considered routes by adding one more pick up location and check each of extended routes until we get $|\mathbb{C}|$ pick up locations in each route. When we extend routes with l $(1 \leq l \leq |\mathbb{C}|-1)$ pick-up locations, we will apply Lemma 1 to avoid extending those routes whose travel distance is already over the cost threshold; When we check routes with $l+1$ $(1 \leq l \leq |\mathbb{C}|-1)$ pick-up locations, we will apply Lemma 2 to prune possible candidates without computing function $\mathcal{G}$. Specifically, a sketch of our algorithm is shown in Figure 5. A tree structure is used to illustrate the strategy of our algorithm as shown in Figure 4, where it is assumed that there are totally six pick-up locations. The root of tree denotes the empty set of candidate, which is essentially a set of candidates with zero pick-up location. If a route at the level l (i.e., containing l pick-up locations) has a potential driving distance over the cost threshold, we do not need to consider all of its extended routes. For instance, if the route $\mathcal{S} \rightarrow B \rightarrow \mathcal{E}$ on the level 1 is already pruned as its driving distance is over the threshold, there is no need to extend this route (i.e., adding a child to B). Also we may prune some candidate routes on the same level using Lemma 2 and their extended ones as well. For instance, if we conclude the travel distance of route $\mathcal{S} \rightarrow E \rightarrow A \rightarrow \mathcal{E}$ is over the cost threshold and those three values of node B mentioned in Lemma 2 are bigger than those of node A respectively, we can directly prune the route $\mathcal{S} \rightarrow E \rightarrow B \rightarrow \mathcal{E}$ without computing its function $\mathcal{G}$.

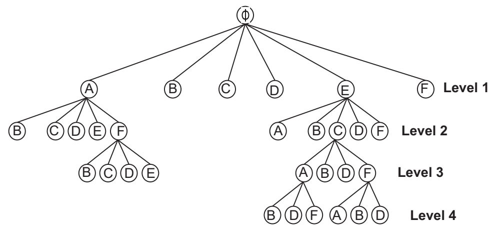


Fig. 4: A Sample of Tree Structures

5 Experimental Results

In this section, we use both real-world and synthetic data to evaluate the efficiency of developed search algorithms and the effectiveness of route recommendations.

5.1 The Experimental Setup

Data and Parameters. In the experiments, we use real-world taxi GPS data that were collected in San Francisco Bay Area. This data set contains around 30-day GPS location traces of approximately 500 taxis in San Francisco Bay Area. For each recorded GPS point, there are four attributes: latitude, longitude, fare identifier and time stamp. The fare identifier could be 1 or 0, where 1 indicates taking passenger(s) and 0 means empty cab. In addition, we generate some synthetic data (i.e., pick-up points and probabilities) for testing the scalability of algorithms. Details about synthetic data will be provided later. We set the parameters for generating pick-up location and probability in Section 3.2 as: $\alpha = 30$ *minutes*, $\tau_1 = 3.0$ *miles* and $\tau_2 = Dist(\mathcal{S} \leftarrow \mathcal{E})/2$, where $Dist(\mathcal{S} \leftarrow \mathcal{E})$ is the driving distance from $\mathcal{S}$ to $\mathcal{E}$.

Experimental Environment. The algorithms were implemented in Python. All the experiments were conducted on a Windows 10 with Intel Core i7 6-Core and 12.00GB RAM. The search time for the optimal driving route is the major metric we use to evaluate the efficiency of our algorithms.

Competing Methods. We compare our search algorithms for RR-RA and DORR problems with two baseline methods. To the best of our knowledge, existing works [7], [8], [9], [10], [11] focus on addressing MSR problem proposed in [6] or other different problems. There is no existing method for solving

```
Input:
        $\mathcal{S}$: the original location
        $\mathcal{E}$: the destination location
        $\mathbb{C}$: a set of pick-up locations
        $\mathbb{P}$: a set of probabilities associated with $\mathbb{C}$
        $\mathbb{T}_{\mathbb{C}}$: a set of pick-up events associated with $\mathbb{C}$
        $TC$: a cost threshold
Output:
        The optimal route from $\mathcal{S}$ to $\mathcal{E}$ for a driver.
1.      for $l = 1 : \mathbb{C} - 1$
2.          Enumerate all routes with $l$ pick-up locations by adding one pick-up location to remaining
            candidates with $l - 1$ pick-up locations; Avoid considering some routes with $l$ pick-up
            locations by applying lemma 1
3.          Prune routes with $l$ pick-up locations by applying Lemma 2
4.      end for
5.      Compute the function $\mathcal{G}$ for all remaining routes and prune those ones
        with travel distance over $TC$
6.      Compute the business success for all remaining routes
7.      Sort them and pick the optimal one with the biggest business sucess
```

Fig. 5: The Algorithm for DORR

either RR-RA or DORR problem. Therefore, the two baselines that we compare to are straight-forward methods for searching optimal route of RR-RA and DORR problems. The idea of baseline method for RR-RA is to directly compute potential driving distance for all candidate routes and find the optimal one with minimum value. The idea of baseline algorithm for DORR is directly computing the business success and cost for each possible candidate route and picking the one with maximum business success and cost lower than the cost constraint TC. All acronyms of evaluated algorithms are given in Table 1.

TABLE 1: Acronyms of Competing Methods

SF-RR-RA:	The baseline search algorithm for RR-RA
Pr-RR-RA:	Our pruning-based search algorithm for RR-RA
SF-DORR:	The baseline search algorithm for DORR
Pr-DORR:	Our pruning-based search algorithm for DORR

5.2 Efficiency of Recommendations

5.2.1 Performances of RR-RA

In this subsection, we show the efficiency of different algorithms for RR-RA problem. We randomly pick some taxi cabs that are empty around 4pm on Friday from our historical GPS data and compute the optimal route for each of them based on its location $PoCab$. To demonstrate the efficiency of our algorithm Pr-RR-RA, we compare its performance with that of baseline method SF-RR-RA.

As shown in Section 3.2, the number of pick-up locations depends on the position and time of an empty cab that requests recommendation service. For each selected empty cab, we first use the method in Section 3.2 to generate the pick-up locations and estimate their pick-up probabilities, then estimate $\tilde{D}_i$ for each pick-up location, and finally compute the optimal route with different algorithms and record their running time. After recording the running time of each algorithm for all cabs, we compute the average of running time for the same number of pick-up points for individual algorithm. In Table 2, we show the average running time for three different numbers of pick-up locations. The running time shown here does not include the time for generating pick-up locations, their probabilities and $\tilde{D}_i$ for both algorithms. From Table 2, we can find that our algorithm Pr-RR-RA outperforms the baseline method significantly. Note that in this set of results we set the max number of pick-up locations of driving route as $\mathcal{L} = 4$.

TABLE 2: Comparisons of Computational Time (Sec) ($\mathcal{L} = 4$)

	SF-RR-RA	Pr-RR-RA
	10 Pick-up Points	
Computational Time	3.19	1.81
	15 Pick-up Points	
Computational Time	6.54	3.08
	20 Pick-up Points	
Computational Time	8.95	4.93

$\mathcal{L}$ is an important parameter that affects computation complexity. Thus we demonstrate the efficiency of different algorithms with different value of $\mathcal{L}$. Specifically, we repeat the same experiments as the above with different values of $\mathcal{L}$. In Table 3, we show the results with 10 pick-up points. The results demonstrate that the computational time of both algorithms generally increases with the increase of $\mathcal{L}$ and our algorithm consistently outperforms the baseline one.

TABLE 3: Computational Time (Sec) with Different $\mathcal{L}$

	10 Pick-up Points	
	SF-RR-RA	Pr-RR-RA
$\mathcal{L} = 4$	3.19	1.81
$\mathcal{L} = 6$	4.38	2.12
$\mathcal{L} = 8$	7.23	4.02

For Pr-RR-RA algorithm, pruning plays an important role to save computational time. We show the average percentage of candidates that are pruned by our algorithm with different values of $\mathcal{L}$ and different numbers of pick-up points in Table 4. As can be seen, a large portion of candidates are pruned without computing EDD value in our algorithm.

TABLE 4: A Summary of Pruning Percentage

10 Pick-up Points			
	$\mathcal{L} = 4$	$\mathcal{L} = 6$	$\mathcal{L} = 8$
Pr-RR-RA	39.7%	41.8%	48.5%
15 Pick-up Points			
	$\mathcal{L} = 4$	$\mathcal{L} = 6$	$\mathcal{L} = 8$
Pr-RR-RA	42.1%	45.3%	51.2%
20 Pick-up Points			
	$\mathcal{L} = 4$	$\mathcal{L} = 6$	$\mathcal{L} = 8$
Pr-RR-RA	41.7%	44.3%	50.5%

To examine the scalability of our algorithm for a large number of pick-up points, we randomly generate potential pick-up points within a specified area, the pick-up probability and $\tilde{D}_i$ associated with each pick-up point. In total, we have 3 synthetic data sets with 50, 100 and 150 pick-up points, respectively. Also we randomly generate 10 positions of empty cab within the same area. For each synthetic data set, we compute and recommend the optimal route for each position of empty cab, and record the computational time of both algorithms. Note that we use Euclidean distance instead of driving distance to measure the traveling distance between pick-up points. In total, we produce 10 recommendations for each data set. We take the average of computational time of these 10 recommendations and show the results in Table 5, where we set parameter $\mathcal{L}$ as different values for the three data sets. As can be seen, our algorithm could scale well with more pick-up points and have more efficiency improvement against the baseline method as the number of pick-up points increases.

TABLE 5: Computational Time (Sec) on Synthetic Data

	SF-RR-RA	Pr-RR-RA
50 Pick-up Points, $\mathcal{L} = 4$		
Computational Time	44.27	21.09
100 Pick-up Points , $\mathcal{L} = 6$		
Computational Time	203.97	82.76
200 Pick-up Points , $\mathcal{L} = 8$		
Computational Time	487.33	158.47

5.2.2 Performances of DORR

In this section, we show the efficiency of different algorithms for DORR problem. We randomly pick some taxi cabs that are empty and far away from their common operation areas around 4pm on Friday from our historical GPS data, and then compute the optimal route for each of them. For each selected cab, we use its current location around 4pm as the original location and the central of its common operation area as the destination location. To demonstrate the efficiency of our algorithm in Figure 5, we compare its performance with that of baseline algorithm SF-DORR.

The number of generated pick-up points depends on original and destination locations. This number may vary among all selected cabs. For each of them, we first generate the pick-up locations and estimate their pick-up probabilities, then we compute the optimal route with different algorithms and record their running time. After recording the running time of each algorithm for all cabs, we take the average of running time with the same number of pick-up points for individual algorithm, and finally show the results in Table 6. Note that the running time shown here excludes the time for clustering of pick up points and probability estimation that is also a part of online search but is the same for all algorithms compared here. By comparing with the baseline method, we can see that our algorithm could consistently save significant computation time over different numbers of pick-up points. Also the computation cost of baseline method increases much faster than that of our algorithm when the number of pick-up points increases.

TABLE 6: Computational Time (Sec) with Different Numbers of Pick-up Point

	SF-DORR	Pr-DORR
6 Pick-up Points		
Computational Time	1.37	0.76
7 Pick-up Points		
Computational Time	7.09	2.06
8 Pick-up Points		
Computational Time	43.85	7.93

To demonstrate the pruning effect, we compute the average percentage of pruned candidates at different steps of our algorithm in Table 7, where we use the same experimental setting as in Table 6. As can be seen, while both Lemma 1 and Lemma 2 enable us to prune many candidates, Lemma 1 leads to more pruning percentage than Lemma 2 does.

TABLE 7: A Summary of Pruning Percentage

Pruning by Lemma 1	Pruning by Lemma 2
6 Pick-up Points	
60.4%	18.7%
7 Pick-up Points	
63.9%	19.6%
8 Pick-up Points	
65.5%	20.9%

The cost constraint (i.e., TC) may affect the efficiency because a lower (higher) value of TC leads to more (fewer) candidates pruned in advance based on Lemma 1 and Lemma 2. To show this effect, we repeat the same experiments as the above yet with different values of TC and show the results in Table 8, where the number of pick-up points is 8. In general, the computation time decreases as the value of TC increases for our algorithm. For real practice, we may need to set up this cost constraint parameter based on the empirical experience and possible input from end users. We may obtain different optimal driving routes with different values of TC with the same other inputs.

TABLE 8: Computational Time (Sec) with Different Values of TC

Values of TC	50 Miles	40 Miles	30 Miles
Pr-DORR	7.93	6.87	5.69

As we mentioned above, the value of TC affects the pruning effect of algorithm. To further demonstrate this, we show the pruning percentage versus different values of TC in Table 9. From Table 9, we could see higher values of TC lead to more percentage of candidates pruned, and consequently less computation time for our algorithm.

5.3 Effectiveness of Recommendations

Other than the computational advantage that we showed above, we examine the effectiveness of our route recommenda-

TABLE 9: A Summary of Pruning Percentage

Pruning by Lemma 1	Pruning by Lemma 2
8 Pick-up Points, $TC = 8$ Miles	
65.5%	20.9%
8 Pick-up Points, $TC = 7$ Miles	
67.7%	21.3%
8 Pick-up Points, $TC = 6$ Miles	
70.3%	23.6%

tions. The purpose of this examination is to demonstrate that the recommended driving routes could help taxi drivers pick up customers. While the ideal way is to conduct field experiments (a.k.a., A/B testing), it is very difficult to collaborate with taxi drivers or taxi companies to run such experiments. Usually, as the first step, offline empirical tests are conducted using classical machine learning leave-out validation methods on historical data, and we follow this approach in this paper. The basic idea of leave-out method is to split historical GPS data into training and testing sets along the time dimension, obtain pick-up points and probabilities using training data, compute the optimal routes for empty cabs in testing data, and compare recommended optimal routes with the routes actually taken by drivers in testing data. Specifically, we use 90% GPS data of each taxi cab as training data and hold remaining 10% GPS data for testing.

5.3.1 Performances of RR-RA

In this experiment, we extract raw drop off locations from testing data, and compute recommended route by treating each drop off location as the origin of an empty cab (i.e., $PoCab$), where we set the parameter $\mathcal{L}$ as 4. In testing data, we have the route actually taken by cab starting from each selected drop off location. We then compare each recommended route with the corresponding actually-taken route by a cab. The actually-taken route starting from an raw drop off location may be longer than the corresponding recommended route. To make them comparable, we cut the actually-taken route to the same length as the corresponding recommended one. Consequently, we measure similarity between each pair of routes (i.e., the recommended route and the cut actually-taken route) based on dynamic time warping algorithm in [44]. In total, there are $42,080$ pairs of route. Because the dynamic time warping algorithm for measuring trajectory similarity is very time-consumption, we randomly sample 5% of pairs for our evaluation. In total we have obtained $2,104$ pairs of route. For each actually-taken route, we know whether there is a pick-up event or not within the cut length based on GPS logs in testing data. Based on this indicator, we split all samples of similarity between each pair of routes into two groups: Group 1 including all similarity samples that has the indicator indicating there is a pick-up event within the really-taken route, and Group 2 including all similarity samples that has the indicator indicating there is no pick-up event. Finally, to get the quantitative measurement of effectiveness, we use statistical t-test to examine the difference between Group 1 and Group 2. There are 1046 similarity samples in Group 1 and its mean is 0.69. On the contrary, there are 1058 similarity samples in Group 2 and its mean is 0.23. The p-value of t-test is less than 0.01, which indicates that the similarity between each pair of routes with actual pick-up event (i.e., Group 1) is significantly bigger than that without actual pick-up event (i.e., Group 2). From these results, we can see that there is a strong correlation between the similarity and the binary indicator. This suggests that recommended routes are very close to the corresponding actually-taken routes that include pick-up events, and that those actually-taken routes that do not include pick-up events are very different from the corresponding recommended routes.

5.3.2 Performances of DORR

We conduct the similar effectiveness evaluation for DORR problem. However, unlike the evaluation of RR-RA, we need to extract drop-off locations, each of which is far away from the common operation area of the corresponding cab. In total, we obtain 890 drop-off locations that belong to the destination-constrained scenario. We consider each selected drop-off location as original location (i.e., $\mathcal{S}$), and the center of the corresponding common operation area as destination location (i.e., $\mathcal{E}$). We search the optimal route to recommend for each selected drop-off location. In total, we also obtain 890 recommended optimal routes. For each selected drop-off location, we obtain the actually-taken route that starts from the drop-off location from testing data. In total, we initially get 890 pairs of recommended route and actually-taken route. But, some actually-taken routes may end at different locations than the corresponding destination locations (i.e., common operation area). To make a fair comparison between each pair of routes, we filter out those actually-taken routes that do not share the same destination location as the corresponding recommended route. After this filtering, we obtain 803 pairs of recommended route and actually-taken route. We use the same method as used in Section 5.3.1 to compare the similarity between two routes of a pair. For each pair, we could obtain a binary variable indicating whether there is a pick-up event or not along the actually-taken route from testing data. Similarly we split the similarity samples into two groups: Group 1 including all similarity samples where there is a pick-up event within the actually-taken route, and Group 2 including all similarity samples where there is no pick-up event. We also use statistical t-test to examine the difference between Group 1 and Group 2. There are 397 similarity samples in Group 1 and its mean is 0.63. On the contrary, there are 406 similarity samples in Group 2 and its mean is 0.28. The p-value of t-test is less than 0.01. In summary, these results suggest that recommended routes are very close to actually-taken routes that include pick-up events, and those actually-taken routes that do not include pick-up events are very different from the corresponding recommended routes.

5.4 Practical Value of Route Recommendation

To further demonstrate the practical value of route recommendation, we estimate the business improvement of our recommendations that is measured as empty rate (ER). ER is defined as the ratio of driving distance without passenger. We can compute ER using historical GPS data for each taxi driver. The lower value of ER means better business profit of a taxi driver. In our data set, the average ER is 55% over all drivers. 10% drivers have ER lower than 0.24, which means that they carry passenger(s) and earn money for 76% of their driving. 10% drivers have ER over 0.7. The goal of our evaluation is to use a simulation method to examine whether

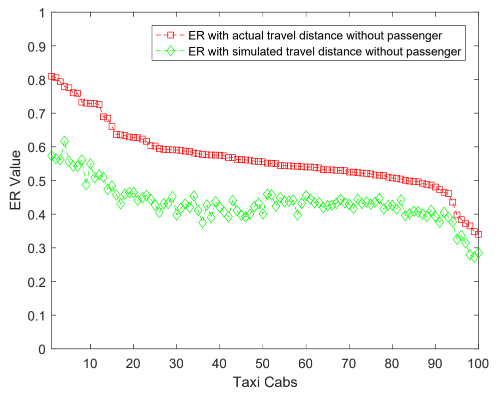


Fig. 6: The Visualization of ER_i and ER_i^S

we can significantly decrease ERs of taxi drivers with our route recommendations.

Specifically, we randomly select 100 taxi cabs from testing data set. For each taxi cab, we extract all driving routes without passenger from 10% of historical GPS data in testing data. For each of these routes, there are a drop-off location at the beginning and one pick-up location in the end. In total, we get $8,030$ routes, i.e., $8,030$ pairs of drop-off and pick-up locations. We use each drop-off location as an input to generate recommended route. Among these $8,030$ pairs, we identify 165 of them as corresponding to DORR problem. For the remaining $7,865$ ones, we consider them as corresponding to RR-RA problem. We use different recommendation methods to search optimal routes for these two groups. For each drop-off location in RR-RA group, assuming the recommended optimal route is taken, we estimate the driving distance before picking up next passenger(s) as the EDD value. This estimated EDD value is considered as the simulated driving distance before picking up next passenger(s) when our recommended route is taken. It is expected that this simulated driving distance is equal to or lower than the actual driving distance prior to pick-up event. For each drop-off location in DORR group, we similarly estimate the simulated driving distance without passenger from S to E when our recommended route is taken. Finally, for each of 100 taxi cabs, we compute two ER values: one (denoted as ER_i) is obtained with the observed actual driving distance without passenger; the other one (denoted as ER_i^S) is obtained with the simulated driving distance without passenger. We then further compute the decreasing percentage of ER value as $DP_i^{ER} = (ER_i - ER_i^S)/ER_i$ $(1 \leq i \leq 100)$. The average of $\{DP_i^{ER}\}$ is 0.18, which means our recommendations could potentially decrease their empty rate of driving by 18%. The t-test also shows that DP_i^{ER} is significantly bigger than 0 with p-value less than 0.01. Moreover, we visualize the values of ER_i and ER_i^S $(1 \leq i \leq 100)$ in Figure 6, where we sort all 100 taxi cabs according to ER_i and align them along x axis. From this figure, we can see that our route recommendations lead to different levels of ER decrease (i.e., $ER_i - ER_i^S$) for different drivers. The higher ER_i is, the more decrease of ER is achieved. All these results indicate that our route recommendations could decrease ER and improve business performance when they are accepted by taxi drivers.

6 CONCLUSIONS AND DISCUSSIONS

In this paper, we introduced a Route Recommendation with Relaxed Assumptions (RR-RA) problem and a Destination-Oriented Route Recommendation (DORR) problem by exploiting massive GPS log, both of which aim to recommend routes to taxi drivers when there is no active pick-up request. The recommended routes can proactively navigate drivers to potential passengers improving their business performance. The package of methods for both RR-RA and DORR problems provides a complete solution for route recommendations: when taxi drivers encounter the destination-oriented scenario, they can obtain route recommendations via the solutions for DORR problem; otherwise, they could get route recommendations with the solutions for RR-RA problem. To generate route recommendations in an efficient way, we developed two smart algorithms for RR-RA and DORR problems, respectively. Both algorithms work based on the identified important monotone property of evaluation functions for RR-RA and DORR problems. Our evaluation results with large-scale GPS data demonstrate the improved efficiency of proposed recommendation algorithms, and the practical value of our route recommendations.

Our developed solutions have potential applications for other transportation problems. The first one is carpool service, which has been provided for many internet-based transportation companies such as Uber, Waze, Takescoop. In the setting of carpool service, many drivers usually have fixed origin (e.g.,home) and destination (e.g., work place) and expect to pick up passengers on the way from origin to destination. How to recommend optimal routes for carpool drivers is an interesting and practically important problem. This routing problem can be essentially formulated as DORR problem, where the objective is to find an optimal driving route that could maximum carpooling probability under given driving distance and direction constraints. The approaches developed for DORR problem can be used for solving this carpool routing problem. The second one is searching for parking lot problem. Searching for parking lot is currently a challenging problem in many big cities (such as LA in the U.S.). Our solutions for RR-RA problem can be used for solving this problem. We can replace the pick-up location and pick-up probability with the parking spot location and spot available probability, respectively. The objective of RR-RA in this setting is then to search an optimal route that leads to the minimum driving distance before finding a parking spot. The recommendation methods and algorithms we developed for RR-RA problem could be used for solving this parking lot search problem. Due to the lack of real-world data, we can not study these two route recommendation problems in this paper.

Finally, we would like to discuss some the perspectives about using the developed route recommendation solution for future autonomous vehicles equipped with sophisticated systems of perception (e.g., camera, LIDAR). With the future autonomous taxi vehicles, computing the optimal driving routes for empty cabs will be more critically needed due to the lack of a driver. Our methods for both RR-RA and DORR problems could provide a good solution to automatically navigate empty driverless taxi cabs. Such automatic navigation could greatly enhance business performance of driverless cabs, save energy and even improve the efficiency of overall transportation systems in an urban area. However, we also realize that there will be new challenges to address the problem of searching optimal driving routes for empty driverless cabs with sophisticated perception systems. More signals about the potential passengers, connected taxi networks, and traffic may be captured by the perception systems of future autonomous vehicles. These signals could be analyzed and utilized for computing the optimal driving routes for empty driverless cabs. In this circumstance, searching optimal routes in the two scenarios of RR-RA and DORR problems (studied in this paper) will be more complicated because additional input signals (e.g., the volume of crowd) captured via the perception systems need to be considered, and consequently new methods and computing algorithms will need to be developed.

Acknowledgment

This research was partially supported by the National Science Foundation (NSF) via grant numbers 1814771 and 1700263.

References

[1] F. Ricci, “Mobile recommender systems,” International Journal of Information Technology & Tourism, vol. 12, no. 3, pp. 205–231, 2011.

[2] G. Abowd, C. Atkeson, and et al, “Cyber-guide: A mobile context-aware tour guide,” Wireless Networks, vol. 3(5), pp. 421–433, 1997.

[3] O. Averjanova, F. Ricci, and Q. N. Nguyen, “Map-based interaction with a conversational mobile recommender system,” in Proceedings of The Conf on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008.

[4] F. Cena, L. Console, and et al, “Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide,” AI Communications, vol. 19(4), pp. 369–384, 2006.

[5] H. van der Heijden, G. Kotsis, and R. Kronsteiner, “Mobile recommendation systems for decision making ’on the go’,” in ICMB, 2005.

[6] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. J. Pazzani, “An energy-efficient mobile recommender system,” in Proceedings of the 16th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, 2010, pp. 899–908.

[7] J. Huang, X. Huangfu, H. Sun, H. Li, P. Zhao, H. Cheng, and Q. Song, “Backward path growth for efficient mobile sequential recommendation,” IEEE Transactions on Knowledge and Data Engineering, vol. 27(1), 2015.

[8] S. Ma, Y. Zheng, and O. Wolfson, “T-share: A large-scale dynamic taxi ridesharing service,” in Proceedings of IEEE 29th International Conference on Data Engineering, 2013, pp. 410–421.

[9] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, “Where to find my next passenger,” in Proceedings of the 13th international conference on Ubiquitous computing, 2011, pp. 109–118.

[10] R. Wang, C.-Y. Chow, Y. Lyu, V. C. S. Lee, S. Kwong, Y. Li, and J. Zeng, “Taxirec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines,” IEEE Transactions on Knowledge and Data Engineering, vol. 30(3), 2018.

[11] Z. Ye, K. Xiao, Y. Ge, and Y. Deng, “Applying simulated annealing and parallel computing to the mobile sequential recommendation,” IEEE Transactions on Knowledge and Data Engineering, vol. forthcoming, 2018.

[12] RecSysworkshop, “http://pema2011.cs.ucl.ac.uk/.”

[13] RecSysworkshoplocalrec, “https://recsys.acm.org/recsys15/localrec/.”

[14] UberSpot, “newsroom.uber.com/us-washington/enabling-seamless-pickups-through-color-coding/.”

[15] UberBeacon, “https://newsroom.uber.com/beacon/.”

[16] LyftAmp, “https://help.lyft.com/hc/en-us/articles/236093888-amp.”

[17] K. Mueffelmann, “https://www.financierworldwide.com/ubers-privacy-violations-a-cautionary-tale-for-others/.wqzdnojytaq.”

[18] Z. Tufekci and B. King, “https://www.nytimes.com/2014/12/08/$opinion/we-cant-trust-uber.html_r = 0$.”

[19] M. Kenteris, D. Gavalas, and D. Economou, “Electronic mobile guides: a survey.” Personal and Ubiquitous Computing, vol. 15, no. 1, pp. 97–111, 2011.

[20] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, “Cyberguide: A mobile context-aware tour guide,” Baltzer/ACM Wireless Networks, vol. 3, 1997.

[21] M. Setten, S. Pokraev, and J. Koolwaaij, “Context-aware recommendations in the mobile tourist application compass,” in Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 2004.

[22] G. Adomavicius and A. Tuzhilin, Context-Aware Recommender Systems, Chapter 7 in Handbook on Recommender System,. Springer, 2011.

[23] K. Cheverst, N. Davies, K. Mitchell, and A. Friday, “Developing a context-aware electronic tourist guide: Some issues and experiences,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2000, pp. 17–24.

[24] L. Ardissono, A. G. Petrone, G. Segnan, and P. Torasso, “Ubiquitous user assistance in a tourist information server,” in Adaptive Hypermedia. Springer-Verlag, 2002, pp. 14–23.

[25] D. Carolis, I. Mazzotta, N. Novielli, and V. Silvestri, “Using common sense in providing personalized recommendations in the tourism domain,” in Proceedings of Workshop on Context-Aware Recommender Systems, New York, 2009.

[26] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, “Incorporating contextual information in recommender systems using a multidimensional approach,” ACM Transactions on Information Systems, vol. 23, no. 1, pp. 103–145, 2005.

[27] K. Bauman and A. Tuzhilin, “Discovering contextual information from user reviews for recommendation purposes,” in Conference on Information Systems and Technology, Nashville, TN, 2016.

[28] C. F. Console, L. Gena, C. Goy, A. Levi, G. Modeo, and T. I, “Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide,” AI Communications, vol. 19, no. 4, pp. 369–384, 2006.

[29] M.-H. Park, J.-H. Hong, and S.-B. Cho, “Location-based recommendation system using bayesian user’s preference model in mobile devices,” in Proceedings of the 4th International Conference on ubiquitous Intelligence and Computing, HongKong,China, 2007.

[30] K. Partridge and B. Price, “Enhancing mobile recommender systems with activity inference,” in Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization, Trento, Italy, 2009.

[31] W. Woerndl, C. Schueller, and R. Wojtech, “A hybrid recommender system for context-aware recommendations of mobile applications,” in Proceedings of the IEEE International Conference on Data Engineering Workshop, 2007.

[32] A. Kamar, “Mobile tourist guide (m-toguide),” in Project Final Report. IST-2001-36004, 2003.

[33] W. Woerndl, J. Huebner, R. Bader, and D. G. Vico, “A model for proactivity in mobile, context-aware recommender systems,” in Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 273–276.

[34] D. Jannach and K. Hegelich, “A case study on the effectiveness of recommendations in the mobile internet.” in Proceedings of the 2009 ACM Conference on Recommender Systems, 2009, pp. 205–208.

[35] Y. Zhang, B. Li, R. Krishnan, and S. Liu, “Learning from the offline trace: A case study of the taxi industry,” in Proceedings of International Conference on Information Systems, 2015.

[36] S. Liu, S. Wang, C. Liu, and R. Krishnan, “Understanding taxi drivers’ routing choices from spatial and social traces,” Transportation Research Part B: Methodological, vol. 9(2), pp. 200–209, 2015.

[37] T. V. Le, S. Liu, H. C. Lau, and R. Krishnan, “A quantitative analysis of decision process in social groups using human trajectories,” in Proceedings of the international conference on Autonomous agents and multi-agent systems, 2014, pp. 1425–1426.

[38] Y. Ge, H. Xiong, C. Liu, and Z. Zhou, “A taxi driving fraud detection system,” in Proceedings of the 11th IEEE International Conference on Data Mining, 2011, pp. 181–190.

[39] S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan, “Vait: A visual analytics system for metropolitan transportation,” IEEE Transactions on Intelligent Transportation Systems, vol. 14(4), pp. 1586–1596, 2013.

[40] D. Sacharidis, K. Patroumpas, M. Terrovitis, V. Kantere, M. Potamias, K. Mouratidis, and T. K. Sellis, “On-line discovery of hot motion paths,” in Proceedings of the International Conference on Extending Database Technology, 2008, pp. 392–403.

[41] S. Rogers and P. Langley, “Personalized driving route recommendations,” in Proceedings of the American Association of Arti Intelligence Workshop on Recommender Systems, 1998, pp. 96–100.

[42] Y. Wang, F. Currim, and S. Ram, “Deep learning for bus passenger demand prediction using big data,” in Proceedings of the Workshop on Information Technology and Systems, 2016.

[43] M. Pinedo, Scheduling Theory, Algorithms, and Systems, 4th ed. Springer, 2016.

[44] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou, “An effectiveness study on trajectory similarity measures,” in Proceedings of the Twenty-Fourth Australasian Database Conference, 2013, pp. 13–22.

Huayu Li received her B.E. degree in Electronic Engineering and Information Science from the University of Science and Technology of China (USTC) in 2011 and Ph.D. in Computer Science from The University of North Carolina at Charlotte in 2018. She is currently a Research Scientist at Facebook. Her research interests include data mining, machine learning and recommender systems.

Yong Ge received his Ph.D. in Information Technology from Rutgers, The State University of New Jersey in 2013, the M.S. degree in Signal and Information Processing from the University of Science and Technology of China (USTC) in 2008, and the B.E. degree in Information Engineering from Xi’an Jiao Tong University in 2005. He is currently an Assistant Professor at University of Arizona. His research interests include data mining, machine learning and recommender systems.

Alexander Tuzhilin received the PhD degree in computer science from the Courant Institute of Mathematical Sciences, New York University (NYU). He is a professor of information systems and the NEC faculty fellow in the Stern School of Business, NYU. His current research interests include data mining, personalization, recommender systems, and CRM. He has served on the editorial boards of the IEEE Transactions on Knowledge and Data Engineering, the Data Mining and Knowledge Discovery Journal, the INFORMS Journal on Computing (as an area editor), the Electronic Commerce Research Journal, and the Journal of the Association of Information Systems.