Smart recommendation by mining large-scale GPS traces

3 authors:

Shiyou Qian
Shanghai Jiao Tong University
8 PUBLICATIONS   **10** CITATIONS

SEE PROFILE

Yanmin Zhu
Shanghai Jiao Tong University
158 PUBLICATIONS   **1,356** CITATIONS

SEE PROFILE

Minglu Li
Shanghai Jiao Tong University
520 PUBLICATIONS   **3,545** CITATIONS

SEE PROFILE

Smart Recommendation by Mining Large-scale GPS Traces

Shiyou Qian, Yanmin Zhu*, and Minglu Li

Shanghai Jiao Tong University; Shanghai Key Lab of Scalable Computing and Systems; * Corresponding Author

{qshiyou, yzhu, mlli}@sjtu.edu.cn

Abstract—**Recommending good driving paths is valuable to taxi drivers for reducing unnecessary waste in fuel and increasing revenue. Driving only according to personal experience may lead to poor performance. With the availability of large-scale GPS traces collected from urban taxis, we have the curiosity about whether we can discover the hidden knowledge in the trace data for smart driving recommendation. This paper focuses on developing a smart recommender system based on mining large-scale GPS trace datasets from a large number of urban taxis. However, such the trace datasets are in nature complex, large-scale, and dynamic, which makes mining the datasets particularly challenging. We first extract vehicular mobility pattern from the large-scale GPS trace datasets. Then, the optimal driving process is modeled as a Markov Decision Process (MDP). Solving the MDP problem results in the optimal driving strategy that gives smart recommendation for taxi drivers. In essence, the most rewarding driving paths can be derived in the long run. We have conducted extensive trace driven simulations and conclusive results show that our recommendation algorithm can successfully find good driving paths and outperforms other alternative algorithms.**

I. INTRODUCTION

Advanced data acquisition techniques have been increasingly utilized in mobile environment for data collection. A probe taxi system is representative for ubiquitous data collection, which uses moving taxis as sensors. The large-scale available trace data collected through such systems provide us with more opportunities to discover useful transportation patterns. These patterns can in turn provide decision-making support for some real-time applications, such as optimal route recommendation for taxi drivers.

Recommendation systems aim at addressing information overloaded problems by identifying personal interests and providing personalized services. This problem has drawn great attention from researchers and academics [1–6]. The research focuses of past were mainly on tourist guide. The work in [1] builds a tour guide prototypes by utilizing user's current location and a history of past locations. However, these algorithms are not suitable for large-scale traces in transportation field. In [7], optimal routes are computed by limiting the length of routes in order to decrease computation complexity. This method has many limitations. First, optimal routes computed by such method may not be overall optimal because the computation is made by considering short-term reward, rather than long-term reward. Second, some experienced taxi drivers are likely to wait at some locations during particular time periods in reality, which is not taken into consideration.

In this paper, we present our study focused on pattern extraction from large-scale trace data which were collected by a probe taxi system in Shanghai, China. Based on the extracted patterns, we develop a smart recommendation system which presents taxi drivers optimal driving routes. When dealing with optimal route recommendation, there are two challenges: i) how to depict the dynamic nature of the problem and ii) how to decrease computation complexity. To tackle the first challenge, useful transportation patterns need to be extracted from the trace data. After the extraction of these patterns, we utilize Markov Decision Process (MDP) to model the taxi optimal route problem. As for the second challenge, computation complexity can be decreased by dynamic programming based on the MDP model. We utilize policy iteration method to compute optimal route in our experiments.

Experiments were extensively conducted on real trace data. The results show that our proposed MDP method outperforms LCP method and RS (Random Select) method with a significant margin. Moreover, we demonstrate that computation complexity of our method is theoretically much lower than other two methods. Our main contributions are:

- Based on large-scale data analasis, we define the optimal drive route selection problem to maximize long-run revenue.
- We employ the MDP framework to compute the optimal policy for route selection.
- Extensive trace driven simulations have been performed and comparitive study has been conducted.

The rest of the paper is organized as follows. Section II formulates the problem. Section III presents system model. Section IV describes data analysis. Section V illustrates performance evaluation results. We conclude the paper in section VI.

II. PROBLEM FORMULATION

In this section, we first give an illustration of the problem and then present the definition of Mobile Optimal Route (MOR) problem.

For taxi drivers, the places where passengers get on taxis are called pick-up points. When a taxi is vacant, the driver needs to get next passenger as quick as possible. For this purpose, the driver should select a place as a potential pick-up point. What kind of strategies would the driver like to take when there are many candidate pick-up points? Selecting by experiences may be of value for experienced drivers, but may
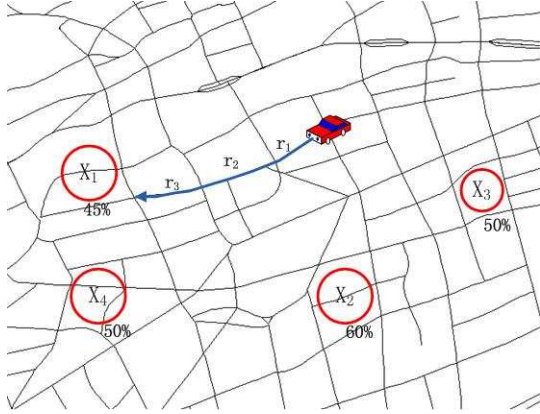
Fig. 1. An illustration of route selection

(a) Distribution of driving distances

(b) Distribution of occupancy rates

Fig. 2. Distribution of daily driving distances and occupancy rates for taxi drivers

lead to poor performance and is not suitable for freshmen. Choosing randomly is simple, but not optimal. Is there a smart strategy for taxi drivers?
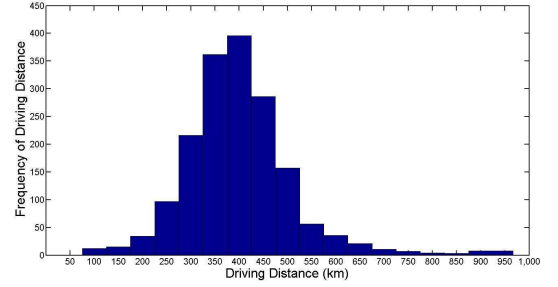
Statistics generated from the analysis of large-scale trace data in our experiments have shown that there exist pick-up points where taxi driver are more likely to take passengers. The degree of likeliness is called success probabilities. These transportation patterns are useful for optimal route recommendation.

An illustration of the problem we are going to address is shown in Fig. 1. There are four potential pick-up points X_1, X_2, X_3 and X_4 with their corresponding success probabilities 45%, 60%, 50% and 50%, respectively. If the driver makes decision only in terms of success probabilities of pick-up points, then X_2 is chosen as the target. However, this strategy neglects the success probabilities of the routes and does not take into account 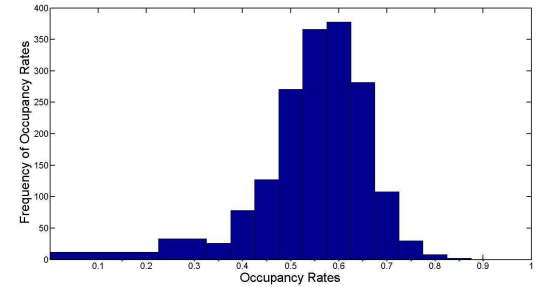the possibility that the driver may fail at X_2. In other words, the driver should consider long-term revenue rather than short-term one. When the driver selects a potential pick-up point as his driving target, many driving routes are available for him to choose. In practice, route with minimal driving time cost can be chosen. When the number of pick-up points is large, there are a large number of candidate routes. How to select an optimal route from the candidate routes is important and challenging.

From data analysis we also find that some taxi drivers have longer daily driving distances(larger than 550 km) and higher occupancy(larger than 0.7) rate. This is shown in Fig. 2. These high-performance drivers knows the time-dependent distribution of pick-up points. We extract these useful transportation patterns and recommend optimal driving routes for unexperienced drivers.

Assume that the set of pick-up points is finite, denoted by $X = \{X_1, X_2, \cdots, X_N\}$. The set of success probabilities for each pick-up point, $P = \{P(X_1), P(X_2), \cdots, P(X_N)\}$, is available, where $0 \leq P(X_i) \leq 1$ is assumed to be independently distributed. The success probabilities for road segments is denoted by $P(r_i)(1 \leq i \leq l)$. The set of choices that a driver can make at a pick-up point is defined by $A = \{Waiting, Moving\ to\ one\ of\ next\ pick-up\ points\}$. The number of choices is dependent on the current pick-up point. Since the number of pick-up points is finite, the transition probabilities between pick-up points when a choice is made can be obtained, denoted by $P_{ij} = Pr(P(X_j|X_i, a))$. A choice made at a pick-up point has a cost C_{ij} and an expected reward R_{ij}; the Potential Expected Revenue(PER) can be computed by a function, denoted by $F(P, P_{ij}, C_{ij}, R_{ij})$, when making a choice.

Based on the definitions and notations described above, the Mobile Optimal Route (MOR) problem can be formally defined as:

Definition 1: MOR Problem

Given: A set of potential pick-up points X with $|X| = N$, a set of success probabilities for pick-up points $P(X_i)$, success probabilities for road segments $P(r_i)$, transition probabilities P_{ij}, costs C_{ij}, expected rewards R_{ij}, and a taxi driver who needs to choose an optimal route.

Objective: Finding an optimal route $\mathcal{R}$ for the taxi driver. The goal is to maximize the PER.

The computational complexity of MOR for BR (Brute Force) method was proven in [7]:

Lemma 1: Given a set of pick-up points X, where $|X| = N$, and $Cox(F) = 1$, the complexity of searching an optimal driving route is $O(N!)$.

III. SYSTEM MODEL

In this section, we first present probability analysis and then utilize Markov Decision process as our problem model.

A. Probability Analysis

As mentioned in Section II, taxi drivers should take into consideration success probabilities of pick-up points, success probabilities of road segments, driving costs, and expected

reward when selecting potential pick-up points. Suppose that the driver select X_k as the next potential pick-up point. Let S be the event that a driver take a passenger along a route toward X_k or waiting at X_k for a while. Assume that a route is connected by l road segments, then we have

$$S = S_1 + S_2 + \cdots + S_l + S_{l+1} \tag{1}$$

where S_i for $i = 1, 2, \cdots, l$ is the event that the drive takes a passenger at road segment r_i, and S_{l+1} is the event that happens at the potential pick-up point. Let p_i be the success probabilities at each road segment for $i = 1, 2, \cdots, l$ and $P(X_k)$ be the success probability at the potential pick-up point X_k for $i = l + 1$. Then

$$P(S_i) = \begin{cases} p_1 & i = 1 \\ p_i \prod_{j=1}^{i-1}(1 - p_j) & i = 2, 3, \cdots, l \\ P(X_k) \prod_{j=1}^{l}(1 - p_j) & i = l + 1 \end{cases} \tag{2}$$

From Equation (2), the event probability that a driver takes a passenger along the route toward X_k or waiting for a while at the pick-up point X_k is calculated

$$
\begin{aligned}
P(S) &= 1 - P(\overline{S_1 + S_2 + \cdots + S_l + S_{l+1}}) \\
&= 1 - (1 - P(X_k))\prod_{i=1}^{l}(1 - p_i)
\end{aligned} \tag{3}
$$

Let t_i be the driving time from the start point to road segment r_i and T_d be the driving time along the route. Then we have

$$E[T_d|S] = \sum_{i=1}^{l} t_i P(T_d = t_i|S) \tag{4}$$

$P(T_d = t_i|S)$ is given by

$$P(T_d = t_i|S) = \begin{cases} P(S_i)/P(S) & i = 1, 2, \cdots, l-1 \\ \frac{P(S_l) + P(X_k)}{P(S)} & i = l \end{cases} \tag{5}$$

Let W be the event that the driver waits at pick-up point X_k. The probability for event W is

$$P(W) = \prod_{i=1}^{l}(1 - p_i) \tag{6}$$

In order to simplify the computation of driving time, we set a fix value t_w for waiting time at the pick-up point. Then the total expected time is given by

$$
\begin{aligned}
E[T|S] &= E[T_d|S] + E[T_w|S] \\
&= \frac{\sum_{i=1}^{l} t_i P(S_i) + t_l P(S_{l+1}) + P(W)t_w}{P(S)}
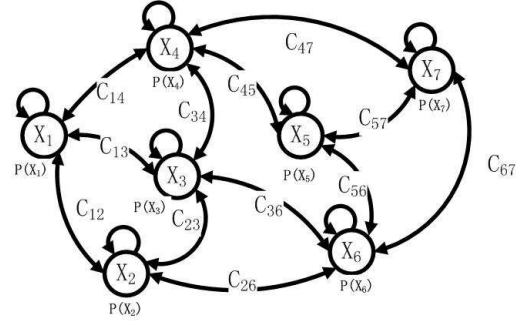\end{aligned} \tag{7}
$$

Fig. 3. An MDP model describing the optimal route problem

Let R_d and R_w be the reward obtained along the route and at the pick-up point, respectively. For the same reason of simplicity, we set a fixed value for R_d. Then the expected reward $E[R|S]$ is

$$E[R|S] = \sum_{i=1}^{l} P(S_i)R_d + P(W)R_w \tag{8}$$

B. MDP Model

In some situations where sequential decision-making is needed under uncertainty, such as optimal route selection, it is appropriate to deploy Markov Decision Process (MDP) model. Bellman et al. first studied the MDP in his seminal work [8].

MDP is defined by a four-tuple $(X, A, P(y|x, a), R(x, a))$, where X is a set of states, A is a set of actions, $P(y|x, a)$ is the probability of transitioning from state $x \in X$ to state $y \in X$ when action $a \in A(x)$ is taken, and $R(x, a)$ is the revenue obtained when action $a \in A(x)$ is taken in state $x \in X$.

The core of MDP is to find an optimal policy for the decision maker: a function that specifies the action $\pi(x)$ that the decision maker will choose when in state x. A policy is said to be optimal if it maximizes the overall expected revenue.

MDP is an appropriate model for our MOR problem:

1) X corresponds with the set of pick-up points.
2) A is the set of available choices at a pick-up point.
3) $P(y|x, a)$ is the transition probability from a specific pick-up point to another point when a choice is made.
4) R(x,a) is Potential Expected Revenue(PER) computed from cost and reward.

The system model for MOR problem is is shown in Fig.3.

Let us denote the state at stage $t \in \{0, 1, \cdots\}$ as x_t and the action chosen at that stage as a_t. Let Π be the set of policies. Each policy, $\pi = \{\pi_t, t = 0, 1, \cdots\}$, is a mapping from X to A, $\pi_t : X \to A$. For a given initial state $x \in X$, the expected total discounted reward (ETDR) for a policy is

$$V^\pi(x) = E\left[\sum_{t=0}^{\infty} \beta^t R(x_t, \pi_t(x_t))|x_0 = x\right] \tag{9}$$

Where $0 < \beta \leq 1$ is the discount factor. The number of policies in the set Π is huge, so the goal of MDP is to find an optimal policy among them that maximizes ETDR. According to Equation (1), the optimal value for a given state $x \in X$ is given by

$$V^*(x) = \sup_{\pi \in \Pi} V^\pi(x) \qquad (10)$$

The corresponding optimal policy yielding that optimal value is denoted by π^*, where

$$V^*(x) = V^{\pi^*}(x), x \in X \qquad (11)$$

In order to find an optimal policy, we denote the optimal one-step reward value for state $x \in X$ in stage i by

$$V_i^*(x) = \sup_{\pi \in \Pi} V_i^\pi(x) \qquad (12)$$

Where $V_i^\pi(x)$ is the one-step reward value for policy π for state x in stage i, which is defined by

$$V_i^*(x) = E\left[\sum_{t=i}^{\infty} \beta^{t-i} R(x_i, \pi_i(x_i))|x_i = x\right] \qquad (13)$$

$i = 0, 1, \cdots$. According to Bellman Optimality Principle, V_i^* can be written recursively as follows

$$V_i^*(x) = \sup_{a \in A(x)} \left\{ R(x,a) + \beta \sum_{y \in X} P(y|x,a)V_{i+1}^*(y) \right\} \qquad (14)$$

For all $x \in X$ and $i = 0, 1, \cdots$.

For infinite-horizon MDP problem, there may not be an optimal policy. The conditions under which an optimal policy exists were proven by Puterman [9].

Lemma 2: If the state space of X is discrete and one of the following conditions is satisfied:

1) $A(x)$ is finite for all $x \in X$;
2) $A(x)$ is compact for all $x \in X$, $R(x,a)$ is continuous on $a \in A(x)$ and $P(y|x,a)$ is continuous on $a \in A(x)$ for $x, y \in X$;
3) $A(x)$ is compact for all $x \in X$, $R(x,a)$ is upper semi-continuous on $a \in A(x)$ and $P(y|x,a)$ is lower semi-continuous on $a \in A(x)$ for $x, y \in X$;

Then there exists an optimal stationary policy.

In the MOR problem, the action set for all $x \in X$ is finite, so there exists an optimal policy for all $x \in X$. The techniques used to determine optimal policy are based on analysis of Equation (14), which utilizes iteration to compute optimal value and determine optimal policy.

We make an assumption that success probabilities maintain constant in a period of an hour. Therefore, an optimal policy obtained from last hour will not be effective at current period. A new optimal policy should be computed according to new success probabilities and costs.
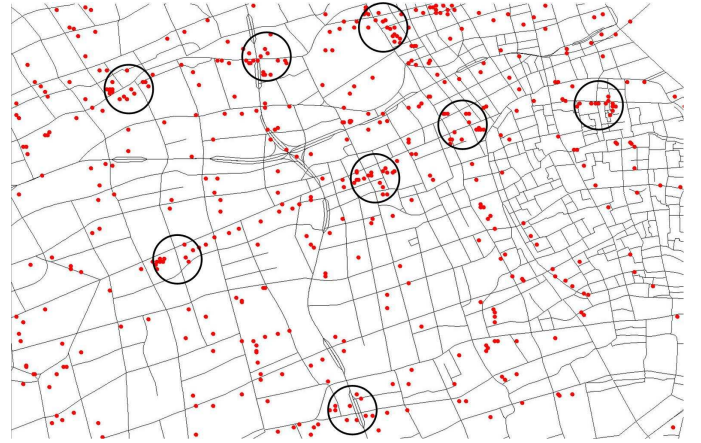

Fig. 4. Distribution of pick-up points for a district between 3:00-4:00 PM on Aug. 15, 2006. Circles represent the potential pick-up points.

IV. ANALYZING TRACES

A. Real Traces

The trace data are the trajectories of approximately 4000 taxis operating around 240 days collected in Shanghai, China by a probe taxi system. The total data size is about 80GB. There are many outliers in the trace data due to many reasons, such as the failure of GPS devices, communication interferences. Trace data that violate any of the following rules are regarded as outliers, and should be filtered out in the experiments.

1) Status field is always ones or zeros.
2) The maximum instantaneous speed is over 120 km/h.
3) The longitude and latitude is out of the range of Shanghai, China.
4) The daily driving distance for a taxi driver is less than 100 km or larger than 1000 km.

B. Patterns Extraction

In this subsection we describe the approaches used to extract useful transportation patterns from the traces data.

1) Pick-up Points: In reality, there always exist some places where more people are likely to take taxis. From the trace data, the points where passengers get on taxis can be extracted. By observing the distribution of these points, we discover that some points are within a certain area. Therefore, proper clustering approaches, such as based on driving distance or Euclidean distance, are applied to organize these points into N clusters. The centroid points of these clusters will be used to represent these pick-up points. Note that the distribution of these pick-up points is changing over time. We also assume that the distribution of pick-up points maintain constant in a period of an hour. Fig. 4 shows the pick-up points distribution between 3:00-4:00 PM on Aug. 15, 2006.

2) Success Probability: After pick-up points have been extracted from the trace data, success probabilities of these pick-up points can be computed. Each pick-up point has a coverage area which is determined by its radius r. We count the number of empty taxis that passed through a pick-up point area, denoted by N_e, and the number of pick-up events that
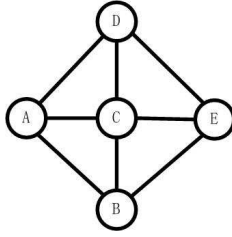
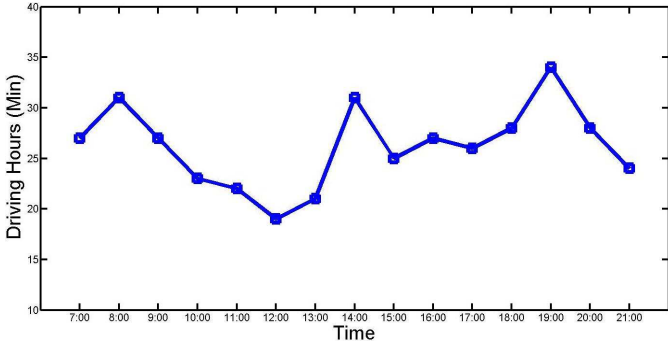Fig. 5. An illustration for choices and transition probability

Fig. 6. Driving time between two pick-up points is dynamic during a day, which is more conformable to the reality than constant distance metric.

Method	Complexity	Description
MDP	$O(MN^2)$	L:length of routes
LCP	$O(N^L)$	M:number of actions
BR	$O(N!)$	N:number of states

As for waiting time, we set a fixed value in order to simplify the model. In the experiments, we set waiting time to 2min, 5min, 8min and 10min to observe its effects. The reward at a pick-up point is also expressed by time. Money is often used as measure of reward in reality. If the duration of a trip is longer, the driver will often earn more money with exception for some unusual conditions, such as traffic jams. In trace data, distribution of trip durations starting at a pick-up point can be calculated.

V. PERFORMANCE EVALUATION

This section presents our evaluation setup, and illustrates the results.

A. Experiment Setting

We used real-world taxi traces in the experiments, which are gathered in Shanghai, China. There are many outliers in the traces. After the elimination of outliers, the traces contain more than 1200 taxis running about 126 days.

All the experiments were conducted on a Windows XP with Intel Core i3 and 4GB RAM. The raw data were processed by programming C# in Microsoft Visual Studio 2008. The MDP optimal policy computation and policies evaluation were carried out by using Markov Decision Process (MDP) Toolbox for Matlab written by Kevin Murphy [10] in Matalab2008a.

B. Compared Algorithms

We propose a new metric incorporating success probability, cost, and reward, namely Potential Expected Revenue(PER). Suppose current pick-up point is X_i, PER for a choice moving to pick-up point X_j is defined as:

$$PER_{ij} = P(S) * (E[R|S] - E[T|S]) \qquad (15)$$

Here $P(S)$ is the expected success probability when driving toward X_j along a route or waiting for a while at X_j, $E[R|S]$ is the expected reward measured in time for selecting X_j as the potential pick-up point, and $E[T|S]$ is the expected driving time cost from X_i to X_j. In terms of the defined metric, we compare three methods used to compute optimal routes:

- MDP is our proposed method.
- LCP is an algorithm used to compute optimal route proposed in [7].
- RS (Random Select) method randomly generates a policy. The PER of RS is the average of 200,000 randomly generated policies.

By the definition of Lemma 1, we know that MOR problem is a combinatorial problem in nature. BS(Brute Force) method for computing the optimal route has to check all the possible routes. In [7], the problem is simplified by constraining the
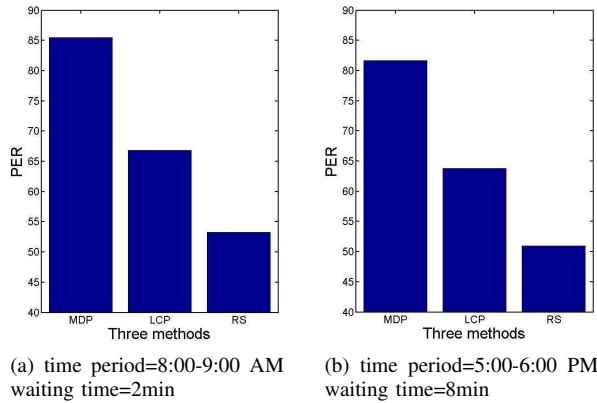
happened at the pick-up point in a given time limit, denoted by N_o. Success probability is obtained by performing the division operation N_o/N_e. Success probability is influenced by radius. With larger radius the success probability is higher. In the experiments, we change the value of radius to observe its effects on success probabilities. Success probabilities are dynamic, changing over time.

Due to the lack of sufficient trace data, success probabilities for road segments can not be extracted directly. However, there are many information that can be used to infer the success probabilities for road segments. Road level is an important information. For example, success probabilities for level 0 or level 1 (main roads or highways) is lower compared with level 2 or level 3 roads. We also assume that roads within proximity have the same success probabilities.

3) Choices and Transition Probabilities: The number of choices depends on the pick-up point. If we consider a pick-up point as node in a graph, the number of choices equals the degree of the node plus one. This is illustrated in Fig. 5. For example, A has three neighbors, so the number of choices at A is four, waiting or moving to one of three neighbors.

The transition probability is deterministic when the choice is made. In other words, the next pick-up point is definitely determined by the choice made at current pick-up point. For example, in Fig. 5, when waiting choice is made at A, then the transition probability $P(X_{i+1} = A|X_i = A, Waiting) = 1$, and other transition probabilities are zero.

4) Costs and Rewards: The driving cost between adjacent pick-up points can be measured by different metrics, such as Euclidean distance, driving distance, or driving time. We utilize driving time as the cost metric, because driving time is dynamic over time, which is more conformable to the reality. Driving time between adjacent pick-up points is computed during different time periods, which is shown in Fig. 6.

(a) time period=8:00-9:00 AM waiting time=2min

(b) time period=5:00-6:00 PM waiting time=8min

Fig. 7.   Comparison of PER

(a) PER w.r.t. radius
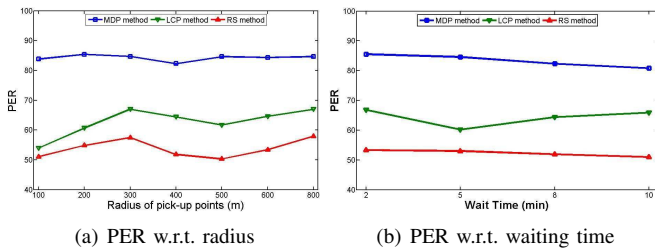
(b) PER w.r.t. waiting time

Fig. 8.   Comparison of PER with regard to different pick-up point radius and waiting time

length of the route in order to decrease computation complexity. Our proposed MDP method utilizes policy iteration to compute the optimal route. The computation complexity of policy iteration algorith was proved to be polynomial in [11]. The theoretical computation complexities are compared in Table I.

C. Evaluation Results

The PER computed by three methods are shown in Fig. 7. MDP method outperforms LCP and RS methods by more than about 25% and 50%, respectively. Experiment results also confirm our analysis mentioned in Section I that the route computed by LCP method is not globally optimal. We get similar results when changing the time period. Due to the space limitation, we do not draw these experiments results.

Success probabilities of pick-up points are influenced by the radius r. We change the value of radius to observe its effects on PER. Our MDP method also outperforms other two methods. The similar result is obtained when we evaluate the impacts of waiting time. This is shown in Fig. 8.

VI. CONCLUSION

Large-scale trace data provide us with opportunities to extract useful transportation patterns and utilize these patterns to improve efficiencies. For this purpose, we develop a smart driving route recommendation system based on transportation patterns extracted from the trace data of taxi drivers. To develop the system, we first formalize the Mobile Optimal Route (MOR) problem and constructed a metric Potential Expected Revenue(PER) for evaluating each candidate routes. We employ Markov Decision Process to model MOR problem

and utilize policy iteration method to compute optimal route. The experiment results show that our method outperforms other two methods in terms of the long-run expected reward. In the experiments, we change different parameters to evaluate the three methods, such as waiting time, radius, and time period.

REFERENCES

[1] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: A mobile context-aware tour guide," *Wireless networks*, vol. 3, no. 5, pp. 421–433, 1997.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, pp. 734–749, 2005.

[3] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[4] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, "Developing a context-aware electronic tourist guide: some issues and experiences," *in Proc. ACM SIGCHI*, pp. 17–24, 2000.

[5] R. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," *in Proc. of the 5th ACM Conference on Digital Libraries*, 1999.

[6] E. Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979.

[7] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," *in Proc. ACM SIGKDD*, pp. 899–908, 2010.

[8] R. E. Bellman, *Dynamic Programming*.   Princeton University Press, 1957.

[9] M. Puterman, *Markov decision processes: discrete stochastic dynamic programming*.   John Wiley & Sons, Inc., 1994.

[10] K. Murphy, "Markov decision process (mdp) toolbox for matlab," 2002. [Online]. Available: http://www.cs.ubc.ca/ murphyk/Software/MDP/mdp.html

[11] C. Papadimitriou and J. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of operations research*, pp. 441–450, 1987.