TaxiHailer: A Situation-Specific Taxi Pick-Up Points Recommendation System

Leyi Song, Chengyu Wang, Xiaoyi Duan, Bing Xiao, Xiao Liu,
Rong Zhang, Xiaofeng He, and Xueqing Gong

Institute for Data Science and Engineering, Software Engineering Institute,
East China Normal University, Shanghai, China
{songleyi,chengyuwang,duanxiaoyi,bingxiao,xiaoliu}@ecnu.cn,
{rzhang,xfhe,xqgong}@sei.ecnu.edu.cn

Abstract. This demonstration presents TaxiHailer, a situation-specific recommendation system for passengers who are eager to find a taxi. Given a query with departure point, destination and time, it recommends pick-up points within a specified distance and ranked by potential waiting time. Unlike existing works, we consider three sets of features to build regression models, as well as Poisson process models for road segment clusters. We evaluate and choose the most proper models for each cluster under different situations. Also, TaxiHailer gives destination-aware recommendations for pick-up points with driving directions. We evaluate our recommendation results based on real GPS datasets.

Keywords: location-based service, taxi, recommendation system.

1 Introduction

The applications of location-based services (LBS) have sprung up in recent years. Based on historical trajectory data, a number of taxi-related recommendation systems and research issues have been proposed. Y. Ge et al.[1] present a mobile recommender system, which has the functionality to recommend a sequence of pick-up points for taxi drivers, in order to maximize the probability of business success. The non-homogeneous Poisson process (NHPP) model has been employed by X. Zheng[2] to describe the behavior of vacant taxis and then to estimate the waiting time of passengers. T-Finder[3] adopts the NHPP model to make road segment recommendations for passengers. It also recommends the top-k parking places and routes to these places for taxi drivers.

It is valuable to make recommendations for passengers using crowd sensing with GPS data of taxi. However, making an accurate recommendation is still a challenge due to many factors, such as the noisy GPS data, the fluctuated weather, the complexity associated with real-world traffic patterns, etc. For example, the NHPP model generally works well for road segments recommendations, but it is not suitable in all situations, such as unpopular roads[2]. Thus, it can achieve higher accuracy through adopting different models for road segments according to the specific situation and adding more features to train models.

S.S. Bhowmick et al. (Eds.): DASFAA 2014, Part II, LNCS 8422, pp. 523–526, 2014.

We develop a situation-specific pick-up points recommendation system for passengers, which takes many factors into account, such as departure point, destination, time, weather and traffic patterns. TaxiHailer distinguishes itself from existing taxi-hailing and recommendation systems in four major ways. First, it focuses on the efficiency of offline processing on large trajectory datasets by taking advantage of the MapReduce framework. Second, we use three sets of features that can influence the waiting time: (i) trajectory-related features, which are statistics calculated based on historical data for road segments, e.g. traffic volume, pick-up rate, etc, (ii) road segment features that describe road properties such as lanes, direction and so on and (iii) additional features including weather conditions. Third, We cluster road segments into groups. For each group, we build regression models and Poisson process model for prediction and deploy the most efficient ones for different situations. Finally, the route to the destination is used to prune the ranked pick-up point candidates, since it is unreasonable to hail a taxi driving in the opposite direction especially on main roads.

2 System Overview

Given a query with specified departure point, destination, time and weather, TaxiHailer gives a ranked list of pick-up points within a specified distance. The points are ranked by the potential waiting time with walking penalty and have been pruned by the direction of route to destination. The models used for prediction are built on historical trajectories collected from taxi GPS devices.

As Figure 1(a) illustrates, TaxiHailer consists of two major parts: (1) offline processing and (2) online computing. The offline modules re-build the waiting time prediction models using recent data periodically. Firstly, in the preprocessing module, we filter the raw GPS data with noises and errors and persist

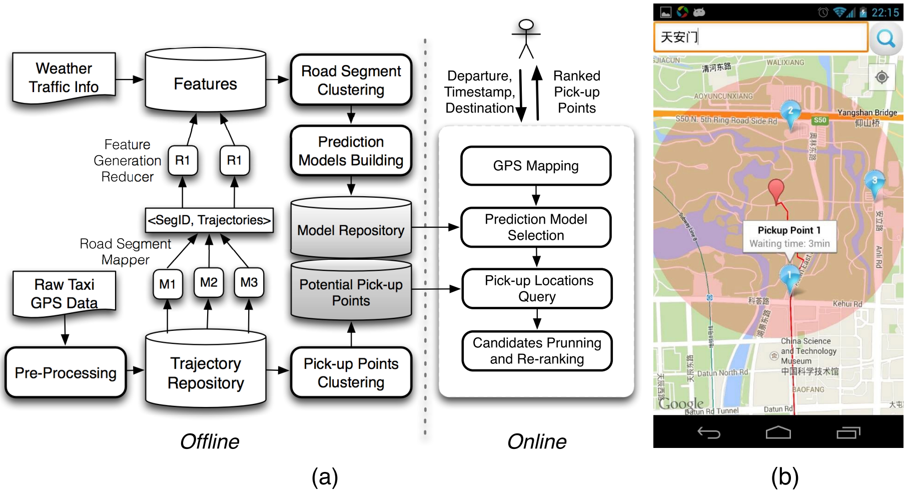

Fig. 1. (a)System Architecture (b)TaxiHailer Screenshot

trajectories into repository. Each taxi trajectory is a collection of GPS points with attributes such as taxi ID, timestamp, taxi state, speed and so on. In the next step, a MapReduce job starts to calculate the statistical features of road segments. We collect additional features, such as weather conditions through web services. Afterwards, road segments are clustered into hundreds of groups to reflect different traffic situations. Besides, to generate potential pick-up points, we clustered historical pick-up points on segments with frequency and distance rules. Then, one point in each cluster is chosen as potential pick-up point for recommendation. Finally, we build regression models and Poisson process models for each road segment cluster under different situations. We choose the best models for each cluster by evaluating using sampled test set.

In the online part, TaxiHailer processes queries and recommends pick-up points within the specified distance. A brief query processing flow is shown as follows. Firstly, the origin of user query is mapped to the road network. Then, we can select the prediction model according to the query context. Next, a collection of result candidates can be fetched by distance limitation. Afterwards, we use the route towards the destination to prune the candidate set, and finally, we re-rank the pruned candidates and return them with predicted waiting time.

3 Challenges and Approaches

In this section, we discuss our implementation details coping with some major challenges in TaxiHailer.

- **Error and Noise Filtering**
  Raw GPS data are never perfectly accurate and contain noise and outlier data points due to sensor noise and other factors. In TaxiHailer, we correct part of inaccurate or noise data from raw GPS data by path inference. For outliers and recording errors, we remove them by setting rules on GPS record attributes. For example, a GPS point should be removed if the distance to the previous point is longer than the possible driving distance in the time interval.
- **Efficient Offline Trajectory Processing**
  Tens of thousands of taxis drive on the roads and generate billions of GPS points per month in big cities, e.g. there are more than 60,000 taxis in Beijing. Huge volume of data accumulated in several months makes accurate predictions possible. Due to the large scale of taxi GPS datasets, we need an efficient mechanism to perform calculation. We implement Map-Reduce flows for the generation of trajectory statistical features. Briefly, the Map tasks map the taxi points to road segments in parallel. The Reduce tasks calculate all statistics we need for the road segments.
- **Situation-Specific Model Building**
  The waiting time for the same road segment may differ in different situations, such as weather, time period. Also, social activities can cause traffic jams, leading to a longer waiting time there. Thus, before model building, we separate anomalous situations from the statistics. Besides, we divide time

into hours and weekdays/weekends/holidays. In practice, we first train set of models for each road segment group. Then, the most proper waiting time prediction model for different situations can be selected by periodically evaluation. We generate sufficient test sets which are sampled on the roads and calculated from the historical trajectory data. In this way, TaxiHailer can yield different recommendations in different situations.

- **Destination-Aware Pick-up Points Recommendation**
  The driving direction is rarely considered in previous taxi recommendation services. However, the volumes of taxis driving on opposite side of the same road are quite different in most cases. It is undesirable to get recommended taxis that drive in the other side. As we generate potential pick-up points including their directions, destination awareness is included in our query algorithm. When the candidate pick-up points are retrieved, we firstly prune them by the direction of planned route to destination. The items in final list are ranked by waiting time plus their walking penalty.

With all the approaches mentioned above, TaxiHailer can effectively select pick-up points for passengers according to destinations and specific queries.

4 Demonstration Scenario

We will demonstrate TaxiHailer and showcase its pick-up points recommendation service. The demonstration is based on taxi GPS datasets in Beijing and Shanghai. More specifically, our demonstration includes two parts shown below:

- **TaxiHailer backend.** We will show how GPS data are processed and models are built in TaxiHailer. Trajectories, road segments, potential pick-up points will be visualized on the map. For a specific situation, different categories of models chosen for road segments will be labelled in colors. We will also report our evaluation results of prediction models in charts.
- **TaxiHailer application.** We will show a mobile application of TaxiHailer as the screenshot is shown in Figure 1(b). Users can generate two kinds of queries: (i) simulated queries based on historical data and (ii) real-time queries. The application will mark the recommended pick-up points on the map. If the destination is specified, an optimized ranked list will be returned.

Acknowledgements. This work is partially supported by Natural Science Foundation of China (Grant No. 61103039 and 61232002) and the Key Lab Foundation of Wuhan University (Grant No. SKLSE2012-09-16).

References

1. Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M., Pazzani, M.J.: An energy-efficient mobile recommender system. In: KDD, pp. 899–908 (2010)
2. Zheng, X., Liang, X., Xu, K.: Where to wait for a taxi? In: UrbComp, pp. 149–156 (2012)
3. Yuan, N.J., Zheng, Y., Zhang, L., Xie, X.: T-finder: A recommender system for finding passengers and vacant taxis. IEEE Trans. Knowl. Data Eng. 25(10), 2390–2403 (2013)