Mobility Mining for Journey Planning in Rome

Michele Berlingerio[1](✉), Veli Bicer[1], Adi Botea[1], Stefano Braghin[1], Nuno Lopes[1], Riccardo Guidotti[2], and Francesca Pratesi[2]

[1] IBM Research Ireland, Ballsbridge, Ireland
{mberling,velibice,adibotea,stefanob,nuno.lopes}@ie.ibm.com
[2] KDDLab Department of Computer Science, University of Pisa, Pisa, Italy
{guidotti,pratesi}@di.unipi.it

Abstract. We present recent results on integrating private car GPS routines obtained by a Data Mining module. into the PETRA (PErsonal TRansport Advisor) platform. The routines are used as additional "bus lines", available to provide a ride to travelers. We present the effects of querying the planner with and without the routines, which show how Data Mining may help Smarter Cities applications.

1 Introduction

Smart Cities applications are fostering research in many fields including Computer Science and Engineering. Data Mining is used to support applications such as optimization of a public urban transit network [3], event detection [2], and many more. Along these lines, the aim of the PErsonal TRansport Advisor (PETRA) EU FP7 project[1] is to develop an integrated platform to supply urban travelers with smart journey and activity advises, on a multi-modal network, while taking into account uncertainty:delays in time of arrivals, impossibility to board a (full) bus, walking speed, and so on. In this paper, we briefly describe the architecture of the PETRA platform, and present the results obtained by the embedded journey planner on thousands of planning requests, performed with and without the results coming from the Mobility Mining module. We show how, by integrating private transport routines into a public transit network, it is possible to devise better advises, measured both in terms of number of requests satisfied, and in terms of expected time of arrivals.

2 PETRA System Components

Figure 1 shows the diagram of a simplified system architecture for PETRA. We list and describe here the main modules used in this paper.

[1] http://www.petraproject.eu

A. Bifet et al. (Eds.): ECML PKDD 2015, Part III, LNAI 9286, pp. 222–226, 2015.
DOI: 10.1007/978-3-319-23461-8_18

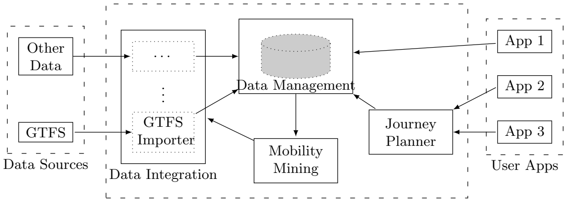

Fig. 1. Simplified PETRA architecture

2.1 Data Management

Handling large volumes of rich and heterogeneous urban data requires a tailored and scalable data management platform, from which we highlight the following modules: i) *data acquisition*, responsible for ingesting heterogeneous urban data; ii) *distributed data storage and indexing*, providing indexes designed for the different formats of data that can be handled by the system (relational, tabular, and graph data), and also their different types (geospatial, textual, etc); iii) *partitioning*, distributing the acquired data across the different nodes of the data storage; iv) *query and searching*, providing a combination of structural query processing and search techniques in order to answer different kinds of queries. The Data Manager (DM) exposes its data to the other PETRA components via a set of APIs, used, for example, by the Journey Planner (JP) to retrieve General Transit Feed Specification (GTFS) data from the DM's internal stored version.

2.2 Mobility Mining

This module fetches GPS data about individual private vehicle trajectories from the DM. We use a data mining process called *mobility profiling* to extract patterns from these traces. This process takes as input the users' trajectories and returns a set of individual *routines* describing their systematic movements [6]. Mobility patterns are expressed as sequences of GPS points with a temporal time stamp that can be exploited as "alternative bus routes" by the JP. These newly introduced routes represent an embedded carpooling service, transparently available in the PETRA application.

2.3 The Multi-modal Journey Planner

We deployed a multi-modal planner taking into account uncertainties related to the expected arrival time of the different modes of transport available in a city. The platform comprising the journey planner provides also functions such as plan execution monitoring, and replanning. The components used in our scenario are the multi-modal JP [4], which is used for the initial planning of journey, and a simulator for plan execution, which is used to monitor the validity of active journey plans. To better perform these tasks, the platform requires updated data. To achieve that, we created a connection between it and the DM, thus deploying the platform in Rome's use case.

3 Case Study

In the Rome's use case, the PETRA platform, from the traveler's perspective, provides journey plans from place A to place B. From the operator perspective, this is done by: importing static and real time urban transport data; merging private routines into the public transport data; computing unvertainty-aware multi-modal advises. We here describe the data used in this paper, how the import step works, and the results obtained with and without private routines.

3.1 Rome Data

The city of Rome, through the public agency Agenzia Mobilità, provides updated open data about its public transport systems. In particular, two main sources of information are offered via its website: i) Rome public transport GTFS, which is a snapshot of the entire public transport network updated every few weeks and ii) Rome public transport real time API. Also, Agenzia Mobilità is gathering a large collection of GPS traces from volunteers? private cars, used by the above described mobility mining module.

3.2 Importing Rome's Data

Importing Rome's data relies on an ad-hoc *data acquisition* module (named RDI, Rome Data Importer), that acts as a bridge between the different kinds of mobility data previously described and the internal DM. RDI performs two sub-tasks: the *daily update* and the *real time update*. The *daily update* consists of discovering *bus stops routines* and enforcing privacy over them. First the RDI transforms the private car routines into sequences of bus stops and combines them as bus lines: each GPS location is mapped to the closest bus stop within a given radius.In order to guarantee car drivers' privacy, the RDI checks if an external attacker could exploit the *bus stops routines* to discover their identity by analysing their vulnerability against the *linking attack* model [5]. Following the methodology in [1], the routines with an identification probability higher than a given *acceptable risk* are transformed into a safer version by removing some bus stops, otherwise they are deleted. Finally all the valid bus stop routines are added to the Rome GTFS data and sent to the DM. Each routine may be used by the JP like any other bus line, even for a portion of the trip. How to make sure the driver of the car can give a ride to the traveler is one of the challenges within the PETRA project. In the *real time update*, the RDI queries the Rome public transport real time API every t minutes, checking for updates (e.g. buses which have been delayed or cancelled) by comparing expected arrival times on the existing GTFS data with real time arrivals. Then it converts possible updates into GTFS format, and sends them to the DM.

3.3 Impact of Routines in Journey Planning

We ran the planning system in two different settings: `NoRo`, in which the planner uses all the public transport data available, but no routines; `Ro`, containing

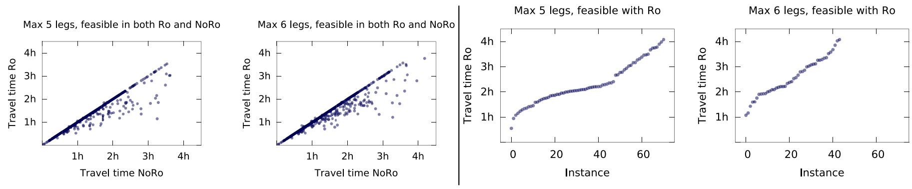

Fig. 2. Impact of routines on travel time.

both routines and public transport data. In each setting, we solved 2,000 queries (instances) with the origins and destinations chosen at random from the logs of the official journey planner of Agenzia Mobilità. In a query, users can set parameters such as the maximum walking time per journey m_w, and the maximum number of legs (i.e., segments) per journey m_l. We set m_w to 20 minutes, the default planner value. Half of the queries have m_l set to 5, and the other half is for $m_l = 6$. The public transport data we used has 8,896 stops and 391 routes. Each route is served by a number of trips, to a total of 39,422 trips per day. The Rome roadmap has 522,529 nodes and 566,400 links. In the GTFS data, we represent routines with a structure similar to public transport data. Each routine introduces a new route and a new trip. We started from 1,205,258 GPS trajectories from 262,657 users. After routine extraction, bus mapping, and anonymization, we ended up with 729 safe mapped routines from 641 users. This increases the number of bus routes to 1,120, for a total number of trips of 40,151.

Figure 2 illustrates the impact of adding routines as an additional mode. At the left, we compare the travel time in the `Ro` and `NoRo` settings. As expected, in a subset of cases, the travel time is the same. On the other hand, all points located below the main diagonal show instances where routines improve the time. In fact, routines can improve both the travel time and the number of legs per journey. The latter has two advantages. First, it makes a trip more convenient to the traveller, as it reduces the number of interchanges. Secondly, it helps increase the set of feasible instances (i.e., instances for which a solution exists). This is important because user-imposed constraints on m_l and m_w can restrict the set of feasible instances. For example, without using routines, in 29.3% of our queries (instances), it is impossible to complete the journey with at most 20 minutes of walking and at most 5 legs in the journey. Charts at the right in Figure 2 show instances that become feasible after adding routines. When m_l is set to 5, routines are part of the returned plan in 17.5% of the instances. Routines increase the percentage of feasible instances by 7.1%, to a total of 77.8%. In 9.6% of the instances, routines improve the travel time, the average savings per trip being equal to 25.5 minutes. When $m_l = 6$, routines become part of the plans in 22.3% of the instances. They increase the percentage of feasible instances from 84.5 to 88.9%. In 14.3% of the instances, routines improve the travel time, the average improvement amounting to 22.05 minutes per trip.

4 Conclusions

We have presented our results obtained by running the PETRA platform on the city of Rome for journey planning. Our results show an increased number of planning instances satisfied thanks to the routines, along with a reduced average expected travel time. Future works include: i) exploiting the uncertainty of the routines for more robust advises; ii) devising the platform for tourism activity planning; iii) extending the mobility mining to crowd patterns.

Acknowledgement. This work has been partially supported by the EC under the FET-Open Project n. FP7-ICT-609042, PETRA.

References

1. Basu, A., et al.: A privacy risk model for trajectory data. In: Zhou, J., Gal-Oz, N., Zhang, J., Gudes, E. (eds.) Trust Management VIII. IFIP AICT, vol. 430, pp. 125–140. Springer, Heidelberg (2014)
2. Berlingerio, M., Calabrese, F., Lorenzo, G.D., Dong, X., Gkoufas, Y., Mavroeidis, D.: Safercity: A system for detecting and analyzing incidents from social media. In: (Demo paper) IEEE 2013 Workshops, pp. 1077–1080 (2013)
3. Berlingerio, M., Calabrese, F., Di Lorenzo, G., Nair, R., Pinelli, F., Sbodio, M.L.: AllAboard: a system for exploring urban mobility and optimizing public transport using cellphone data. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013, Part III. LNCS, vol. 8190, pp. 663–666. Springer, Heidelberg (2013)
4. Botea, A., Nikolova, E., Berlingerio, M.: Multi-modal journey planning in the presence of uncertainty. In: ICAPS (2013)
5. Monreale, A., Andrienko, G.L., Andrienko, N.V., Giannotti, F., Pedreschi, D., Rinzivillo, S., Wrobel, S.: Movement data anonymity through generalization. Transactions on Data Privacy 3(2) (2010)
6. Trasarti, R., Pinelli, F., Nanni, M., Giannotti, F.: Mining mobility user profiles for car pooling. In: KDD 2011, pp. 1190–1198 (2011)