On Recommending Opportunistic Rides

Nicola Bicocchi, Marco Mamei, Andrea Sassi, Franco Zambonelli
University of Modena and Reggio Emilia, Italy
{nicola.bicocchi,marco.mamei,andrea.sassi,franco.zambonelli}@unimore.it

***Abstract*—Research on social and mobile technologies recently provided tools to collect and mine massive amounts of mobility data. Ride sharing is one of the most prominent applications in this area. While a number of research and commercial initiatives already proposed solutions for long-distance journeys, the opportunities provided by modern pervasive systems can be used to promote local, daily ride sharing within the city. We present a set of algorithms to analyse urban mobility traces and to recognise matching rides along similar routes. These rides are amenable for ride sharing recommendations. We validate the proposed methodology using data provided by a large Italian telecom operator. Assuming the full set of considered users are willing to accept 1 Km detours, experimental results on two large cities show that more than 60% of trips could be saved. These results can be used to evaluate the potential of a ride sharing system before its actual deployment and to actually support an opportunistic ride sharing recommender system.**

I. Introduction

The large-scale adoption of smart phones and networking tools produces massive amount of data. This allows us to observe the daily life of people in a previously inconceivable way. A number of works focus on inferring from these data the *mobility* habits of people [1], [2] and can possibly enable innovative forms of ride sharing to reduce the number of circulating vehicles in urban scenarios [3], [4].

While ride sharing for *planned* and *long-distance* journeys have been proposed from some years (e.g., *BlaBlaCar.com*, recent pervasive capabilities might allow ride sharing to support for *short-term, local* journeys within the city. Pervasive technologies and mobile computing, in fact, can be used to automatically recognise matching rides without user interaction. In this direction, in our research we focus on two main aspects: *(i)* the methodology used to identify sharing opportunities by analysing mobility patterns, and *(ii)* the evaluation of an autonomous *recommender system* based on the results of mobility pattern analysis.

As discussed in [5], mobility demand covers many aspects of our life and even though home-work commuting is a relevant part, other daily activities have relevant effects as well. Accordingly, systems designed for reshaping mobility demand need to take into account several aspects of our everyday life including leisure and free time. Because of this, we designed an approach to extract general mobility routines (other than home-work commuting) to support ride sharing opportunities during the whole day of the users.

Some previous research already proposes approaches to extract routines from mobility data. For instance, in [3], [6] authors exploit GPS traces for inferring users' frequent routes or mobility profiles. In [3] some analysis on telecom GSM data has been performed as well. Our methodology exploits Call Detail Records (CDR) collected over the cellular network as the main source of localization and mobility data. Considering that CDR data covers a large population of users, our analysis can support the estimation of ride sharing potential before its actual deployment, and an actually running system in the bootstrap phase when the ride sharing application does not have collected enough GPS points to infer mobility routines.

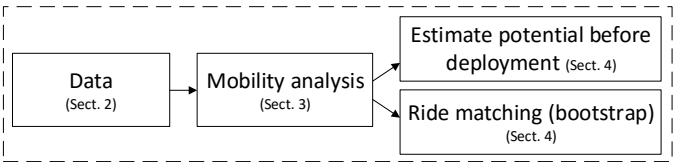


Fig. 1. **The system architecture. The recommender extracts mobility routines from data provided by a telecom operator. Then uses a matching algorithm for identifying ride sharing opportunities.**

While most approaches focus on ride sharing for home-work commutes, and match rides only at departure and arrival locations, we provide two key innovations:

- General mobility routines identification through a probabilistic model running on the whole set of available trips
- En-route travel matching supporting both driver or rider diversions.

While it is difficult to compare results among different cities, our algorithm for en-route pick-up and drop-off notably improves ride sharing opportunities (see Table 3). This paper proposes the design and development of a recommender system for ride sharing structured like the one depicted in Figure 1. More specifically:

- We discuss the mobility dataset used as input (CDR) representing people whereabouts (Section 2).
- We propose a methodology for extracting information from mobility traces both in terms of home-work commutes and general mobility routines (Section 3).
- We propose algorithms to match multiple users' routines thus enabling ride sharing opportunities (Section 4).
- We estimate the amount of rides and kilometres that could be saved by deploying the proposed system. Experiments show that, if users are willing to tolerate up to 1 Km detours, 40-70% of trips and 10-40% of kilometres (depending on the city) could be saved (Section 5)

II. Dataset

We obtained a mobility dataset provided by a major mobile operator. In particular, we made use of anonymised mobility data for about 6 million people living in Piemonte and Lombardia (two italian regions) spanning one month. We focus the analysis in the largest city of each region: Turin and Milan.

TABLE I
THE TABLE SHOWS THE STRUCTURE OF THE DATASET WE USED. USERS SENDING OR RECEIVING CALLS OR TEXT MESSAGES ARE RECORDED AS ONE RECORD COMPRISING THE USER (HASHED) ID, THE MMC (MOBILE COUNTRY CODE), THE TIMESTAMP, THE CODE OF THE CELL TOWER, ITS GEOGRAPHICAL COORDINATES AND COVERAGE RADIUS

User	MMC	Timestamp	Tower	X,Y	Radius
3a	223	7355286	121	(42.2,13.7)	550
3a	223	7355565	128	(42.2,13.8)	400
3a	223	...	...	...	...

Mobility data is represented by Call Detail Records (CDRs) and Mobility Management (MM) procedure messages (i.e., International Mobile Subscriber Identity – IMSI – attach/detach and Location Update). CDRs are continuously collected from mobile networks for billing purposes. A CDR is generated each time a phone sends or receives calls or text messages. The IMSI procedure marks the phone device as attached or detached to the mobile network when the device is switched on or the SIM is inserted.

CDRs and MMs messages are collected from network interfaces through specific sensors. CDR data represent the location of the phone's owner over time approximated to the antennas' radius. Table I shows an exemplary CDR record. Each record is composed by the user id (anonimized), the MCC (Mobile Country Code) representing the country where the SIM card has been activated, the timestamp, the code of the network antenna and its coverage. It is worth mentioning that we do not estimate cells' coverage with Voronoi tessellation. Instead, we represent a cell with a circle (i.e., center and radius) [7]. We used this representation for three principal reasons: *(i)* the dataset we used actually describes cell's coverage with circles. This form efficiently approximates the accepted model with sectored cells of 120-degrees. *(ii)* This representation naturally deals with overlapping cells covering the area of events (overlapping cells are - by definition - avoided by Voronoi tessellation). *(iii)* From a computation perspective, it is easier to deal with cells modeled as circles than polygons.

In Figure 2(a) we show the cumulative density function (CDF) of the average number of CDR events produced by users every day. Indeed, a large portion of users produces a significant amount of CDR events. Specifically, the most active 10% records more than 10 CDR events per day allowing fine-grained mobility tracking. Figure 2(b) illustrates, instead, the CDF of the radius of gyration. It is a simple parameter defining the displacement of the user positions from the centroid. It is given by: $r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - p_{centroid})^2}$ where p_i represents the i^{th} position recorded for the user and $p_{centriod}$ is the center of mass of the user's recorded displacements obtained by: $p_{centroid} = \frac{1}{n}\sum_{i=1}^{n}(p_i)$. It is worth noticing that the first quartile usually associates with sedentary people with $r_g < 5Km$. The most of the distribution (25^{th}-75^{th}) percentiles might be associated with people living in cities because peri-urban areas of major Italian cities frequently have a radius around 15km. Users beyond the 75^{th} percentile are, instead, considered as commuters. Figure 2(c) shows the CDF of the cell radius for the cities under analysis, this implicitly defines users' localization accuracy *if* they would generate

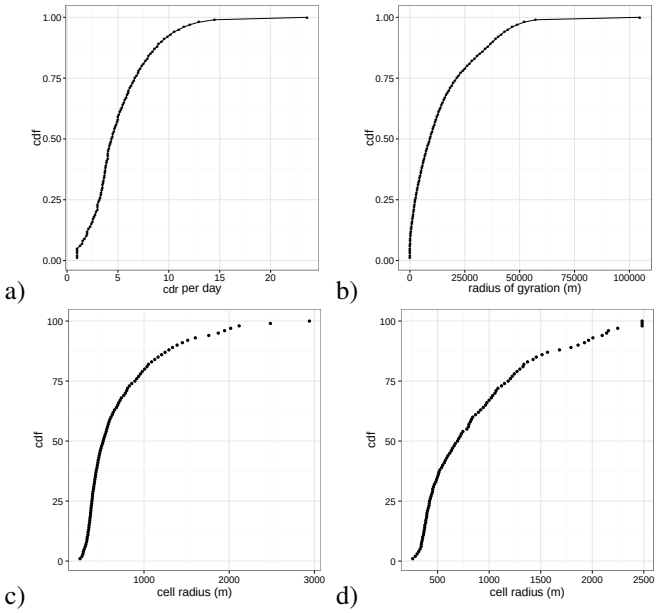


Fig. 2. **CDF of: (a) average number of (daily) CDR. (b) radius of gyration. (c) cell radius. (d) cell radius of CDR actually produced by users.**

CDRs uniformly among cells. Figure 2(d) shows the CDF of the cell radius according to their actual use by the people (i.e., the CDF of the cell radius of the generated CDRs). Considering this latter measure as best reflecting localization accuracy, it is possible to see that half of the localizations have a radius below 700m and 90% are within 1.5Km. Although CDR localization is less accurate than GPS, the error margins are still compatible with acceptable possible detours by drivers and pedestrians. In general, the characteristics of CDR data make it useful to support before-deployment evaluation (i.e., before having a mobile app deployed) and possibly at application's bootstrap phase. After that, GPS traces collected by the ride sharing app will likely outperform CDR data.

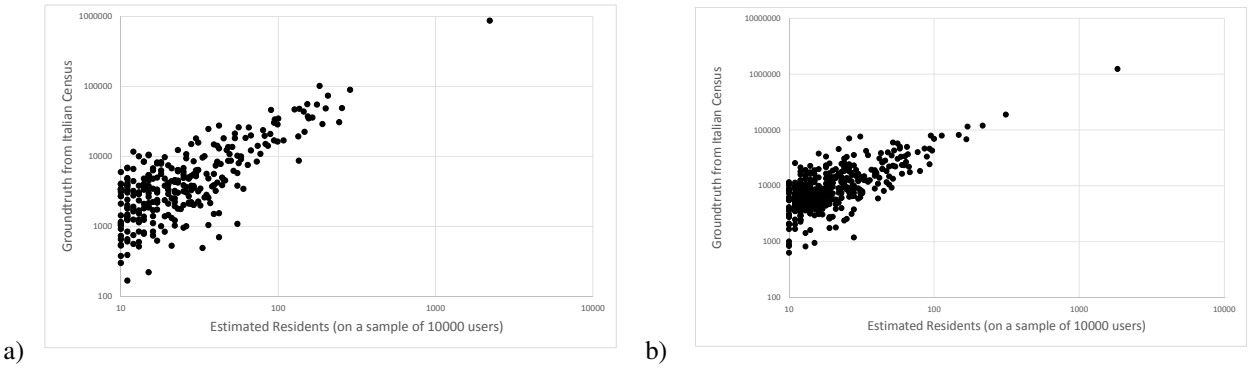
III. MOBILITY ANALYSIS

Mobility analysis is key for autonomous ride sharing system because of its ability of predicting where a user is heading. On the other hand, it also allows the prediction of mobility routines where data is missing (if on a given day, the user does not use his/her phone the system does not have any information about the user's whereabouts. Nevertheless, via pattern analysis, it can infer user's mobility on the basis of past behaviours).

A. Home-Work Routines

In this section, we describe: *(i)* how we extract home and work locations; *(ii)* how we evaluate the precision of the obtained results. For each user, we identify the average commuting hours and her daily routines.

1) Identification: In identifying home-work locations, we adopted a clustering method similar to the one presented in [8], [9], [10] considering all the mobility events of each user (see Figure 3). **(1)** For each user, CDR events are collected. **(2)** Events are then spatially clustered for identifying geographical



a) b)

Fig. 4. **Log-Log plot showing the correlation between the number of residents in different municipalities as measured by our approach and by the national census. (a) In Piemonte. (b) In Lombardia.**

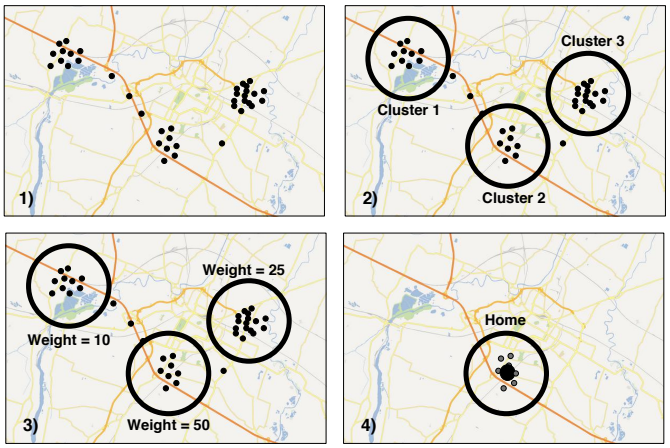


Fig. 3. **Relevant places' identification (i.e., home and work). (1) Selection of CDR events for each user. (2) Spatial clustering of CDR events. (3) CDR events weighing considering both day, time and events nearby. (4) Home and Work identification with the most weighted clusters.**

areas. **(3)** Then, clusters are weighted by taking into account of the number of days in which the user visited them. **(4)** The centers of the most weighted clusters are finally labelled as home and work places. In particular:

Clustering. Due to the design of the mobile network, geographic areas often host multiple cells. As a consequence, events produced around the same place might actually be associated with different cells. Unfortunately, this approach might lead to the misrecognition of relevant places. For example, users calling from specific places might scatter their traces among several cells. Each one of these cells will receive only a portion of events, thus resembling a less frequented place. To avoid this fragmentation process, we spatially aggregated events by making use of an agglomerative algorithm based on geographic distance [11].

We selected this specific algorithm for the following reasons: *(i)* those algorithms (e.g., Spectral Clustering, K-Means) requiring the number of searched clusters are inadequate because CDR events can be spread over large areas and an unpredictable number of clusters. *(ii)* Density-based algorithms (e.g., SOM, DBScan) do not suffer this issue solve but do not naturally limit the clusters' size [11]. Constraining the clusters's size is relevant in this context because the number of cells that can be associated with specific positions is limited to the cells "around" that position. Considering our aim of clustering elements falling within a given radius from a given place, we decided to use agglomerative clustering algorithm.

Weighting. For the sake of weighing the found clusters, we defined two time windows one ranging from 9pm to 6am and the other ranging from 11am to 4pm associated to "home" and "work" respectively.

We used these rather large fixed time window for two main reasons: *(i)* it is very simple, efficient and used in the majority of the state of the art [8], [9] *(ii)* as CDRs are rather sparse, approaches trying to learn user's home and work time from data could lead to overfitting. For each cluster, we also defined a "home-weight" and "work-weight" equal to the number of days in which the user produced one event at least. We reasonably assume that home and work places are candidate to be outliers in our weight distribution because they are visited almost every day.

Thresholding. We then computed mean μ and standard deviation σ of the obtained weight distribution and empirically identified a threshold $th = \mu + \sigma$ separating outliers from the rest of the clusters. Once a cluster is selected, we label the centroid of the cluster as home or work.

2) Validation and Discussion: Accuracy of home place identification has been evaluated by comparison with data from the national statistics office. We selected a sample of 10000 individuals and correlate the density of their homes with resident density from census-based information. Figure 4 shows correlation results for the two regions under study (the log-log scale has been used to reduce changes due to cities of different sizes). Considering a linear model and forcing the intercept to 0, we obtain a best fit: $census = 369.6 \cdot telecom$ and $census = 656.6 \cdot telecom$ (correlation coefficient around 0.9) for Piemonte and Lombardia respectively. Further results on the basis of a limited sample of test users (for whom groundtruth information were available) is reported in [10].

3) Mobility Model: On the basis of the identified home and work places, we can estimate the home-work commuting behavior of individuals. In this section we assume that user mobility can simply be approximated by two daily trips: one from home to work and another from work back to home.

For the home-to-work trip, for each user and for each day we considered the last CDR generated form the home location followed by a CDR from the work location. We average the time of the CDR from the home location to estimate departure time. We average the time of the CDR from the work location to estimate the arrival time. Vice versa, for the work-to-home trip, we average the last CDR time from work as departure time, the first CDR time from home as arrival time. Finally, we compute the actual path (*Route*) that connects the two locations, through a journey planner (i.e., http://www.graphhopper.com) constrained over the OpenStreetMap network.

Overall, each trip is described by a record with the fields: *Source Time*, *Source Location*, *Destination Time*, *Destination Location*, *Route*.

B. General Mobility Routines

As presented in the introduction, there are many routines other than home-work commute that have an impact on the overall demand for mobility, e.g., shopping or leisure-related mobility. Therefore, systems aiming at reshaping mobility demand comprehensively have to take into account also other aspects of our whereabouts. Here, we discuss a topic modeling approach [12] extending home-work commute. These models identify routine characterizing the typical mobility of users.

1) Routines Identification: Following a procedure similar to [13], [14], we first represent users' mobility traces via bags of words (in which each word describes a single movement of a user – disregarding the temporal sequence of these movements). Then, the topic modeling algorithms (i.e., Latent Dirichlet Allocation - LDA) identifies repeating patterns and regularities in those words. Such patterns (called topics) describe users' typical mobility routines.

Bag of Words. For each user and for each day, we consider the sequence of all the users' CDRs. For each pair cdr_i and $cdr_j (j = i+1)$, we assign a trip from the center of the cdr_i cell to the center of the cdr_j cell departing at time t_i and arriving at time t_j (t_i and t_j being the time of the cdr_i, cdr_j respectively). More specifically, we discretized both the location and the time of CDR: *(i)* We partitioned the region in a grid with square cells of 1Km side. Each CDR location is mapped to the corresponding cell, and the cell id is considered as the word location. *(ii)* We considered 24 time slots, one for each hour. As we will describe in the following, the reason for this spatio-temporal discretization, is to create enough *overlaps* in the bag of words to allow the LDA algorithms to detect routines.

The result of this process is a bag of words each comprising two location labels and two time labels. Each word also stores the actual geographic coordinates (without grid-based discretization) associated to that movement so as to precisely localize the CDRs locations, and to compute the distance (taking into account the road network) that is covered during the transition. The resulting *bag of words* is the input data structure for the LDA algorithm (see Figure 5).

LDA - Latent Dirichlet Allocation. LDA is a probabilistic generative model [12] used to cluster documents according to the word patterns (i.e., topics) they contain. LDA has two key advantages compared to other clustering approaches (e.g., DBSCAN): *(i)* LDA is a mixed membership model allowing documents to be represented by multiple topics; *(ii)* Topics are represented as meaningful word distributions that are more easily interpretable [13].

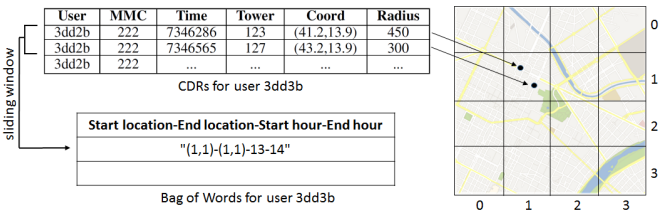


Fig. 5. **Bag of words representation. For each pair cdr_i and $cdr_j (j = i + 1)$ we define a word with the location of cdr_i, the location of cdr_j , and the departing at time t_i and arriving at time t_j (t_i and t_j being the time of the cdr_i, cdr_j respectively).**

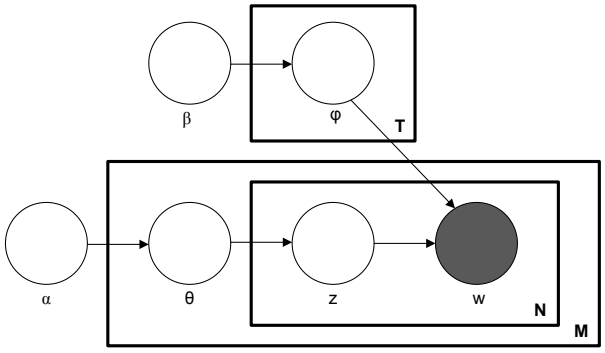


Fig. 6. **The LDA model. w_{ij} are the specific elements in the bag of words. They are the only observable variables in the model. $i \in [1, M]$ represents the document (day). $j \in [1, N]$ represents the j-th word (movement) in the document. z_{ij} is the topic for the j-th word in document i. They are are modelled as multinomial distributions. θ_i is the topic distribution for document (day) i, ϕ_j is the topic distribution for word j. These variables are modelled with a Dirichlet distribution. α and β are the parameters of the uniform Dirichlet priors.**

We emphasize that we are not proposing an extension of the LDA model: we use it in a new way for identifying mobility routines and ride sharing opportunities thereof. LDA is based on the graphical model represented in Figure 6. A word w is the unit of observable data. It represent the movement of a user. *N* words describe a day of the user. Each user is described by *M* documents (i.e., days). Each day is represented as a mixture of topics z. Each topic is represented by the list of all possible words (i.e., movements) associated to the probability $p(w|z)$ (i.e., topics are multinomial distributions over words). Therefore, for each day i, the probability of a word w_{ij} is given by $p(w_{ij}) = \sum_{t=1}^{T} p(w_{ij}|z_{it})p(z_{it})$, where *T* is the number of topics. $p(w_{ij}|z_{it})$ and $p(z_{it})$ are modeled as multinomial distributions. Mixture parameters are modeled with Dirichlet distributions with parameters α and β. In our settings, both parameters were set to 1 representing a uniform distribution. We use Gibbs sampling implemented in Mallet library(*http://mallet.cs.umass.edu*) to set the model parameters. Once the model parameters have been learnt, it is possible to rank z on the basis of $p(d|z)$ (i.e., to extract the topics best describing the routines of a given day).

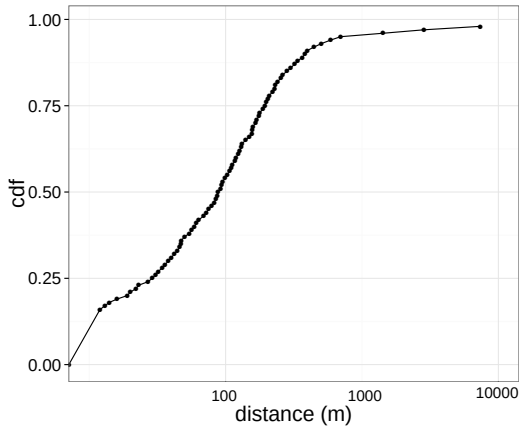


Fig. 7. **LDA evaluation. Y axes shows the cumulative distribution function of the fraction of daily movements that are represented in the top words of the most probable topic., while on X axes is reported the distance (in meters) between the top 2 locations based on topics and home-work locations discovered with the agglomerative clustering approach.**

2) Validation and Discussion: It is difficult to provide sound measures of the accuracy of the extracted topics because ground-truth information is missing (this is an open research question in topic modeling). Therefore, we test if the LDA model *generalizes* the home-work commute described in Section 2. For each user, we computed the home and work locations described in Section 2 with the two most probable locations extracted by LDA associated to events during the day (i.e., work) and during the night (i.e., home). Specifically, we measured the distance between them (see Figure 7). Results show that for more than 90% of the users, the distance between these points is less than 1 km, thus validating our approach.

3) Mobility Model: Once LDA topics have been identified, to create a mobility model of each user, for each day of the week, we computed $p(w|d)$ as $p(w|d) = \sum_z p(w|z)p(z|d)$. This is the probability that a given trip (i.e., word) is going to happen on a given day. This model describes how each user tends to move from multiple locations that include *and* extend home and work places. We generate mobility routines for each user and for each day, in the form: *SourceTime*, *SourceLocation*, *DestinationTime*, *DestinationLocation*, *Route* by considering all the words with a probability grater than a certain threshold, taking place at not-overlapping times. In our experiments we used $p = 0.3$ to identify recurrent routines (i.e., those happening 30% of days). Also in this case, we compute the actual path (*Route*) that connects the two locations on the basis of OpenStreetMap.

IV. Ride Matching and Recommendations

Both previous approaches generate a mobility model for each user in the form of a set of tuples containing: *OriginTime*, *OriginLocation*, *DestinationTime*, *DestinationLocation*, *Route*. This allows to apply the same ride matching to both the previous approaches (i.e., both for home-work commutes and for general routines).

Our ride matching algorithm considers both en-route matching (i.e., pick-up or drop-off location on the path), and both the case in which the driver diverts form the route and the case in which the passenger reaches a meeting location on the driver original route.

The ride matching algorithms basically compares pairs of mobility models to identify whether: *(i)* they take place at the same time (within a certain time window), *(ii)* origin and destination locations of one's mobility are close to the route (points) of the other one (i.e., the trip of the former user is contained in the trip of the latter one). In the case both conditions apply, we consider the two users as a candidate pool. The user with the contained trip becomes the rider, the user with the containing trip becomes the driver. We also considered the capacity of cars by limiting aggregation to 4 users together maximum ($seatCapacity$ parameter). Figure 8(a) illustrates exemplary matching scenarios. Assuming all the trips proceed from left to right, trip B can be merged with trip A (A becomes the driver, B becomes the passenger). Trip A (or even Trip A+B) can be merged with trip C (C becomes the driver, A and B become passengers). Trip D cannot be merged with others. Figure 8(b) details the process of letting driver and rider to actually encounter. In one case (passenger diversion) the passenger walks to the pick-up location. In another case (driver diversion - rightmost figure) the driver diverts to reach the passenger.

Algorithm 1 - Assignment Procedure ($\mathcal{T}$)

1: $\mathcal{P} \leftarrow emptyPoolsCollection$
2: **for** $i \in \mathcal{T}$ **do** *// $\mathcal{T}$ is the set containing the trips*
3: **if** $assigned(i) == FALSE$ **then**
4: **for** $j \in \mathcal{P}$ **do** *// $\mathcal{P}$ is the set containing the pools*
5: $R_j \leftarrow getRidersAssigned(j)$
6: **if** $(R_j < seatCapacity)\ and\ (match(i,j) == TRUE)$ **then**
7: $R_j \leftarrow R_j \cup getUser(i)$
8: $assigned(i) \leftarrow TRUE$
9: **break**
10: **if** $assigned(i) == FALSE$ **then**
11: **for** $k \neq i \in \mathcal{T} : assigned(k) == FALSE$ **do**
12: **if** $match(i,k) == TRUE$ **then**
13: $assigned(i),\ assigned(k) \leftarrow TRUE$
14: $\mathcal{P} \leftarrow \mathcal{P} \cup newPool(i,[k])$
15: **break**
16: **else if** $match(k,i) == TRUE$ **then**
17: $assigned(i),\ assigned(k) \leftarrow TRUE$
18: $\mathcal{P} \leftarrow \mathcal{P} \cup newPool(k,[i])$
19: **break**
20: **if** $assigned(i) == FALSE$ **then**
21: $assigned(i), \leftarrow TRUE$
22: $\mathcal{P} \leftarrow \mathcal{P} \cup newPool(i,[])$
23: **break**
24: **return** $\mathcal{P}$

The actual algorithm to compute matches on the mobility models is described in Algorithm 1. This algorithm gives priority to the already discovered users' pools (lines 4-9). If the available pools do not offer any match, the algorithm looks for any match with the concurrent trips (by the temporal window) not yet aggregated (lines 11-19). If none of the individual trips

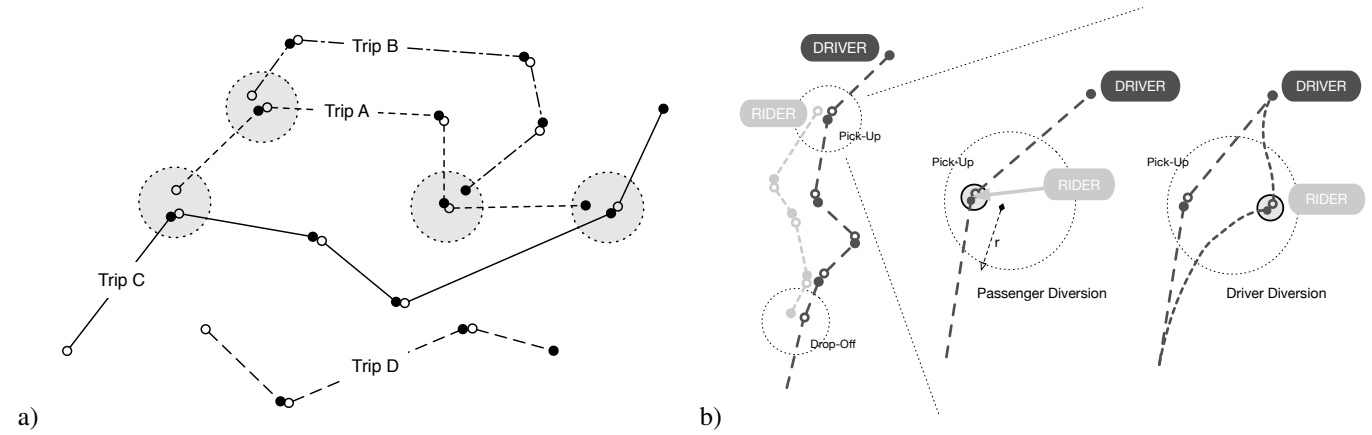


Fig. 8. **(a) Exemplary aggregation strategies for the analyzed trips. Trip B can be merged with Trip A (A becomes the driver, B becomes the passenger). Trip A (or even Trip A+B) can be merged with Trip C (C becomes the driver, A and B become passengers). Trip D cannot be merged with others. (b) An example of rider and driver diversions. The leftmost figure (in b) is an examples of trips being merged. In one case (passenger diversion) the passenger walks to the pick-up location. In another case (driver diversion - rightmost figure) the driver diverts to reach the passenger.**

offers a solution, then the trip is considered as a new pool for following aggregations (line 22). As previously depicted by Figure 1, the exploited methodology points at:

- identifying shared rides (by considering both drivers' and riders' diversions) that can work as a benchmark to evaluate the possible impact of a ride sharing system before its actual deployment (i.e., before the ride sharing application has been installed by a critical mass of users)
- bootstrapping the system when the ride sharing application still does not have collected enough GPS points to infer users' mobility routine (after then bootstrap phase, GPS data will likely outperform CDR measures).

V. Experimental Results

To assess our proposal, we run experiments on two large Italian cities (i.e., Torino and Milano) taking into considerations: *(i)* daily commutes and *(ii)* general mobility routines. We considered both the case in which *(iii)* riders divert from their path to reach a pick-up location, and *(iv)* drivers take a detour in order to collect riders. In each of the four originating cases, we measured the reduction in terms of the amount of trips and km saved (see Figure 9).

In the case of riders' diversions, we tested the matching mechanism (see Algorithm 2) for different values of the distance users were willing to walk (δ) to reach a shared car. For each pair of driver-rider paths, we asserted that the rider should not walk more than δ, and that neither the driver nor rider have to wait more than a temporal window τ at the pick-up meeting point (line 10) and that the expected arrival time of the rider should be within τ to her/his destination time (line 12).

Results are depicted in Figure 9 (a,d). In Torino, the number of trips could be halved by adopting a trip sharing system and accepting to walk at most 500m to reach a shared vehicle. In Milano, the number of trips could be halved by walking around 1000m.

Instead, when drivers are expected to divert in picking-up potential passengers, we have assumed that their likelihood

Algorithm 2 - Matching Mechanism with Rider Diversion match($driverPath, riderPath$)

1: $\tau \leftarrow temporalWindow$
2: $\delta \leftarrow walkingThreshold$
3: $D_P \leftarrow getPathPoints(driverPath)$
4: $R_{OL} \leftarrow getOriginLocation(riderPath)$
5: $R_{DL} \leftarrow getDestinationLocation(riderPath)$
6: $R_{OT} \leftarrow getOriginTime(riderPath)$
7: $R_{DT} \leftarrow getDestinationTime(riderPath)$
8: $match \leftarrow FALSE$
9: **for** $p_i \in D_P$ **do**
10: **if** $(d(p_i, R_{OL}) \leq \delta)$ *and* $(\tau \leq |R_{OT} - getTimeAt(p_i)|)$ **then**
11: **for** $p_j \neq p_i \in D_P$ **do**
12: **if** $(d(p_j, R_{DL}) \leq \delta)$ *and* $(\tau \leq |R_{DT} - getTimeAt(p_j)|)$ **then**
13: $match \leftarrow TRUE$
14: **break**
15: **if** $match == TRUE$ **then**
16: **break**
17: **return** $match$

of sharing is subjected to the impact of the diversions with regard to the length of their original routes, when traveling alone. Thus, in these scenarios we varied the spatial threshold on a ratio between the expected diversion distance over the driver's path distance.

In Algorithm 3, we mined driver-riders matching by considering a driver maximum detour in distance equal to a certain ratio (δ) of her initial trip (D_D) for each rider they pick-up, and another equal detour for each rider they drop-off. Furthermore, to assure savings on driven kilometers, we added a constraint on the total covered distance by the driver while she shares rides: $d(D_{OL}, p_i) + d(p_i, R_{OL}) + d(R_{DL}, p_j) + d(p_j, D_{DL}) \leq D_D$ (line 14). Total actual detour required to the driver in picking up and dropping off riders must be less or equal to her initial trip. Spatio-temporal closeness is evaluated for each passenger at both meeting points (lines 12-14).

Algorithm 3 - Matching Mechanism with Driver Diversion match($driverPath, riderPath$)

$\tau \leftarrow temporalWindow$
$\delta \leftarrow detourRatioThreshold$
$D_P \leftarrow getPathPoints(driverRoute)$
$D_D \leftarrow getPathDistance(driverPath)$
$R_{OL} \leftarrow getOriginLocation(riderPath)$
$R_{DL} \leftarrow getDestinationLocation(riderPath)$
$R_{OT} \leftarrow getOriginTime(riderPath)$
$R_{DT} \leftarrow getDestinationTime(riderPath)$
$R_D \leftarrow getPathDistance(riderPath)$
$match \leftarrow FALSE$
for $p_i \in D_P$ **do**
    if $(d(p_i, R_{OL}) \leq \delta \times D_D)$ *and* $(\tau \leq |R_{OT} - getTimeAt(p_i)|)$ **then**
        for $p_j \neq p_i \in D_P$ **do**
            if $(d(p_j, R_{DL}) \leq \delta \times D_D)$ *and* $(\tau \leq |R_{DT} - getTimeAt(p_j)|)$ *and* $(d(D_{OL}, p_i) + d(p_i, R_{OL}) + d(R_{DL}, p_j) + d(p_j, D_{DL}) \leq D_D)$ **then**
                $match \leftarrow TRUE$
                break
        if $match == TRUE$ **then**
            break
return $match$

In our study, we bypassed the complexity of computing distances constrained by the road topologies by considering spherical distances (computed in $d(p_1, p_2)$ functions through haversine formula) between origin-destination geographic coordinates of the rider and their closest points covered by the driver with her/his route. Nevertheless, final saving computations are done on the actual road topologies. Therefore, the constraint imposed on the total covered distance by the driver did not ensure savings on the total driven kilometers (given the mismatch between the two distances). Accordingly, we introduced a multiplying factor (α) that addresses the issue (i.e., it makes the spherical distance as large as the actual one:

$$\alpha \times [(d(D_{OL}, p_i) + d(p_i, R_{OL}) + d(R_{DL}, p_j) + d(p_j, D_{DL})] \leq D_D$$

In our experiments, best results have been registered with $\alpha = 4$. A lower value of α implied an increase on the amount of saved trips (not circulating cars) and a decrease on the amount of saved kilometers (eventually negative). Results are depicted in Figure 9 (b,e).

It is worth noting that interesting results have been depicted for both the studied cities. The scenario that assumes "walking riders" could reduce the amount of circulating cars more than 50% in Torino, while in Milano it reaches savings of 40%, by accepting to walk at most 1000 meters to get collected by a driver. Drivers detours, instead, produce a less efficacy reduction of 25% on the amount of needed cars in the best case. Thus, we can conclude that the riders' diversions scenario offers a more effective car pooling with regards to the drivers' diversions scenario. However, both scenarios depict a reduction on the amount of needed cars and driven kilometers, to address all the mobility needs of people. This may reduce many issues related to traffic. The results reported by UberPool [15] are aligned with ours.

TABLE II
EXECUTION TIME IN MINUTES OF RIDE MATCHING ALGORITHMS FOR DIFFERENT SPATIAL THRESHOLDS)

Passenger Diversion		Driver Diversion	
S. Threshold	Exe. Time (mins)	S. Threshold	Exe. Time
1400m	180	10%	168
1700m	189	15%	180
2000m	195	20%	190

In the last set of experiments, Figure 9 (c,f), we analyzed the impact of time delays τ. We report the experiments on general routines and passenger diversion (HW-routines and driver diversion produces analogous results). Delays are expressed in hours, as it is the discretisation used in the LDA algorithm. Therefore, for example, $\tau = 0$ means that a ride should happen in the same hour ($\pm 0 \div 59$ minutes) of the original trip, while $\tau = 1$ means that a ride should happen within two hours ($\pm 0 \div 119$ minutes). Results show that there is a 10% gap between $\tau = 0$ and $\tau = 1$. While, the difference gets smaller further increasing $\tau = 1$. We used τ = 1 in this analysis.

Finally, it is worth noticing how specific features of the cities under our evaluation led to significantly different results. For example, in Torino the amount of trips could be halved by asking riders to walk 500 meters while in Milano the same saving is achievable by walking almost 1200 meters. It is likely that the reasons of these different reported behaviors are mainly placed in the different structure of the cities, topology of the road networks, spatial distribution of key locations (i.e., home and work places).

For example, differently from the most of Italian cities having a set of main radial roads culminating in the very city center, Torino has a Roman structure. Indeed, the most relevant roads cross each other at 90 degrees following the traditional Roman *Castrum*. This difference with Milano, which instead has a radial structure, might explain a portion of the observed differences. A better understanding of these factors, might lead to the definition of policies and guidelines to help in redesigning more efficient urban environments for large-scale ride sharing systems.

From the perspective of computational costs, all the experiments have been conducted on a single node octa-core i7-3770@3.40GHz CPU with 24GB DDR3 RAM. Clustering and LDA algorithms can be easily parallelized by dividing users among the cores. Since each user is analyzed in isolation the computation can be evenly divided. Our implementation process 10000 users in 6.5 minutes.

The ride matching algorithm is more computationally intense as in the current version matches all the 75000 trips across each other (currently with only minimal optimizations). Results for different values of spatial thresholds are reported in Table II.

The algorithm has an execution time that grows linearly with the parameter settings. Moreover, since users' routines are rather stable, this analysis can be computed once in a while (e.g., on a monthly basis) so the execution time remains acceptable even for larger number or trips. We are currently improving our algorithms to run on a Apache Spark cluster

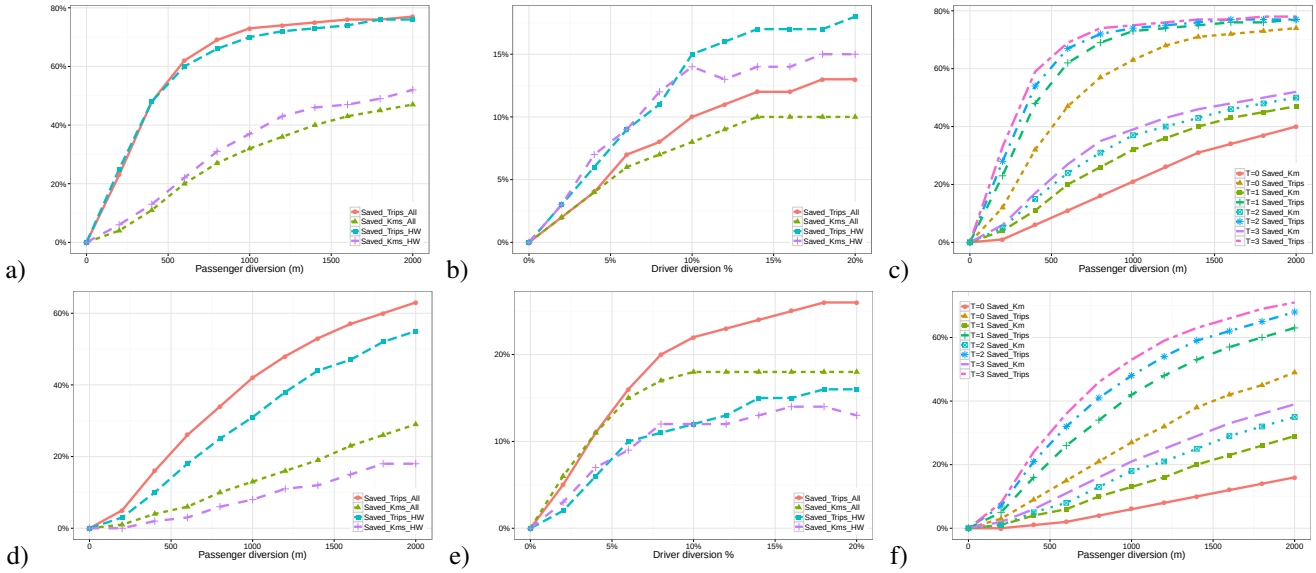


Fig. 9. **Experimental results for both Torino (a,b,c) and Milano (d,e,f). In Torino, with riders diversions (a), the number of circulating vehicles can be reduced more than 50% by joining ride sharing and walking at most 1km to get a lift. The same scenario in Milano (d) offers to save up to the 40% of the needed cars. While, with drivers diversions, both in Torino and Milano (b,e), cars can be reduced up to 15% in Torino and more than 20% in Milano by accepting a maximum diversion (for each rider picked-up) around 10% of the original recurrent trip. Plots (c,f) show the impact of time delays (0 means in the same hour) in the matching process.**

TABLE III
TABLE SUMMARISING RECENT RESEARCH WORKS AIMING AT QUANTIFYING THE BENEFITS OF ADOPTING LARGE-SCALE SHARING SYSTEMS IN VARIOUS GEOGRAPHIC AREAS.

Study	City	Trips saved
Alexander, Gonzales [16]	Boston	38%
Bicocchi, Mamei [17]	Torino	35%
Cici et. al. [18]	Madrid	50%
Santi et. al. [19]	New York	95%
Uber (UberPool) [15]	N/A	32%
Trasarti et. al. [3]	N/A	55%

and take advantage of a parallel computation pipeline for the ride matching algorithm.

VI. Related work

The first applications promoting car pooling and ride sharing have been developed almost 20 years ago. One of the first car pooling service - *Mitfahr-Zentrale* - was deployed in 1998. One of the most popular - *BlaBlaCar* - was launched in 2003. Nowadays, the vast majority of systems propose sharing rides on *planned* and *long* ($> 100km$) trips. The well known BlaBlaCar enables peer-to-peer ride sharing among users, which can get and offer rides with a bid based approach similar to eBay. While this platform has spread around many countries, and it is grounded on a massive user base, it still requires direct inputs from its users to identify matching rides, thus the task of seeking for matching mobility needs and capabilities is not fully automated.

Recently, the popular ride sharing applications Uber (*www.uber.com*) and Lyft (*www.lyft.com*) introduced a feature allowing any users (instead of registered drivers only) to share trips with other strangers. In select cities where Uber's version, UberPool, is available, Uber confirmed to Venturebeat that the service reduces traffic congestion by roughly 55 percent [15].

All these systems require users to explicitly enter requests for rides, and to explicitly enter travel availabilities (e.g., BlaBlaCar) or being notified about nearby requests (e.g., Uber and Lyft). The ride sharing platform we present, instead, *autonomously* looks for sharing opportunities by analyzing recurrent user's routines.

A. Mobility analysis for Ride Sharing

Recently, several works tried to estimate the benefits of large-scale ride sharing platforms in terms of trips, vehicles, or congestions saved. Results have been summarised in Table III and discussed below.

In [16], authors explore the impact of ride sharing adoption on congestion using mobile phone data. They extract average daily origin-destination (OD) trips from mobile phone records and estimate the proportions of these trips made by auto and other non-auto travellers. Next, they match spatially and temporally similar trips, and assume a range of adoption rates for auto and non-auto users, in order to distill ridesharing vehicle trips. Assuming 50% adoption rate, they found: number of vehicles -19%, vehicles mile traveled -11%. They also studied the impact on traffic congestion obtaining a reduction of congested traffic time by 35%. The main difference between this and our work is that they focus on home-work commute only, while our LDA approach extracts also general routines.

In [18] authors analyse users' localizations collected by mobile phones to enable ride sharing among people with overlapped mobility routines on a daily base. The total number of circulating cars could be halved at the expense of 1 Km detours, if people that share the same home and work areas would also share their private cars. Thanks to the approach in

the mobility routine identification provided by the proposed topic model, the work we presented here goes further by identifying both general routines and home-work commutes.

Using graph matching algorithms over New York, in [19] authors found that 95% percent of all taxi trips could be shared, without significantly increasing waiting times. These results are also due to the regular grid topology of New York.

In [3] authors analyse GPS data to extract users' mobility profiles (i.e., frequent paths) via scan statistics and clustering mechanisms. Then they run ride sharing experiments obtaining that 32% of trips could be saved if users a willing to walk for 2.5Km to reach the ride and tolerate up to 1 hour delay. Authors generalize the analysis to GSM data (CDR) reporting larger ride shareability due to inaccuracies in the user localization both in space and time (i.e., inaccuracies soften the spatio-temporal matching constraints). The algorithms we propose for mobility analysis and ride matching are novel with respect to this proposal. For example, we consider both the cases of driver and rider diversion to the pick-up/drop-off point. In general, a direct comparison between the algorithms is difficult as results strongly depend of the data being used and the city plan and mobility routines. However, our results are in line with previous findings.

In our previous works described in [17], [20], we evaluated already the ride sharing potential in Torino. However, in those works we did not taken into account en-route meeting points for riders pick-up and drop-off, but only similar departure and arrival locations were considered. Under that limit, we obtained 35% trips saved for maximum passengers walking threshold of 1 kilometer. Our current approach improves that result by considering en-route meeting points and using the actual roads topology. Moreover, we better compare results deriving from the other ride sharing scenario of drivers detours, and between the ones obtained for home-work commutes and the ones concerning generalised mobility routines.

B. Ride Sharing and Applications

Hall and Qureshi in [21] develop a probabilistic model that evaluates the likelihood that a person will successfully find a ride-match within a set of potential ride matches. Dynamic Ride-sharing (DR) is envisioned as an automated process by which individuals find ride matches on a trip-by-trip basis. They examine the DR concept on both a theoretical basis and on an actual implementation in Los Angeles. Specifically, their paper investigates the likelihood that the user of a DR system would be successful in finding a ride match. This paper shows that dynamic ride sharing is a viable concept. The same authors also report the lack of communication and social norms, among other obstacles, which concur to the low utilization rate of ride sharing support services.

In [22], authors evaluate the issue further, by proposing a set of social enablers for ride sharing. Many efforts in these directions were made by analysing social and psychological aspects of individuals while solicited through persuasive technologies [23] capable of influencing behaviors and coordinating crowd-sourced activities. We believe that these aspects, beside the analysis of incentives and reputation mechanisms [23], influence the practical applicability and success of ride sharing systems relevantly, and we consider to deeper investigate on this issue in our future works.

On this line, the work in [24] considers a combination of intelligent repositioning decisions and dynamic pricing for the improved operation of shared mobility systems. Specifically, they apply incentives mechanisms to a shared bicycle system (but similar approaches could be applied to car as well) to encourage users to park bicycles at nearby under-used stations, thereby reducing the expected cost of repositioning them.

The work in [25] presents network flow technique to systematically develop a long-term many-to-many car pooling model. Long-term car pooling is defined as the sharing of a private vehicle by more than one user who need to reach a destination following a semicommon route between the individuals' points of origin and destination in a specific period. Authors employ a network flow technique to systematically develop a long-term many-to-many car pooling model. The model is formulated as a special integer multiple-commodity network flow problem. Results confirm the usefulness of the model and the heuristic algorithm and that they could be useful in practice.

An interesting application scenario for ride sharing systems is tackled in [26]. This work describes the *last-mile transportation* scenario: the movement of people and goods from a central hub to a final destination, as a challenging topic in implementing eco-friendly transport systems. In this work, they propose and design a system platform for mitigation of vehicle distribution in an electric vehicle (EV) sharing scheme for last-mile transportation. Although they focus on a completely different set of challenges, their approach can be coupled with our sharing mechanism to improve performance.

In [27] authors propose a method which aims to best utilize ride sharing potential while keeping detours below a specific limit. The method specifically targets ride sharing systems on a very large scale and with a high degree of dynamics which are difficult to address using classical approaches known from operations research. For this purpose, the road network is divided into distinct partitions which define the search space for ride matches. They also proposed a novel agent-based approach for match making among drivers. They test the algorithm for taxi sharing in Singapore. The outcome shows that the number of trips could be reduced by 42%.

In [28], authors propose another option to ease traffic problems by exploiting participatory social interactions. They depict a crowd-sourced parking service that collect and update the occupancy data of parking slots in the city. They focus on a different case study, but we think that crowd-sourcing can act as a complementary tool to the automatic process of extracting mobility profiles (i.e., places and routines) we propose. Furthermore, authors, confirms that the expected participation rate is a key factor when designing a sharing system: a system with a lower expected participation rate will place a higher burden in individual participants.

VII. Conclusion

The set of algorithms depicted in our work can support an urban-scale recommender system in the identification of ride sharing opportunities by mining mobility traces collected from

telecom operators. Obtained results can help estimating the potential of a ride sharing system before its actual deployment and in supporting the actual operation of the system especially in the bootstrap phase where other data (e.g., GPS) may be missing. While a number of research works and companies proposed ride sharing services for planned and sporadic travels which cover long distances, our approach makes use of the novel opportunities offered by pervasive technologies to bring ride sharing on everyday trips at an urban-scale, which occur frequently and usually cover short-medium distances. Experimental results showed how the traffic can be decreased and its related issues mitigated thanks to our feasible approach.

This work also showed that the same ride sharing approach might have different results on different cities (Milano and Torino have been taken as examples). This fact might suggest that specific urban features such as road topology and the distribution of residential or working areas impact on large-scale sharing systems. On these basis, in our future work, we plan to analyse other cities for the sake of identifying policies and best practices allowing sharing systems to be readily adopted in urban environments.

REFERENCES

[1] B. Furletti, L. Gabrielli, G. Garofalo, F. Giannotti, L. Milli, M. Nanni, D. Pedreschi, and R. Vivio, "Use of mobile phone data to estimate mobility flows. measuring urban population and inter-city mobility using big data in an integrated approach," in *Proceedings of the Meeting of the Italian Statistical Society*, Cagliari, Italy, 2014.

[2] M. Nanni, R. Trasarti, B. Furletti, L. Gabrielli, P. V. D. Mede, J. D. Bruijn, E. D. Romph, and G. Bruil, "Transportation planning based on gsm traces: A case study on ivory coast," in *Citizen in Sensor Networks*, Springer International Publishing, 2014.

[3] R. Trasarti, F. Pinelli, M. Nanni, and F. Giannotti, "Mining mobility user profiles for car pooling," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego (CA), USA, 2011.

[4] F. Zambonelli, "Toward sociotechnical urban superorganisms," *IEEE Computer*, vol. 45, no. 8, pp. 76–78, 2008.

[5] B. Tuner, *One Planet Mobility: A Journey towards a sustainable future*. WWF Report, 2008.

[6] W. He, D. Li, T. Zhang, L. An, M. Guo, and G. Chen, "Mining regular routes from gps data for ridesharing recommendations," in *Proceedings of ACM SIGKDD International Workshop on Urban Computing UrbComp Í2*, Beijing, China, 2012.

[7] M. Ulm, P. Widhalm, and N. Brandle, "Characterization of mobile phone localization errors with opencellid data," in *IEEE International Conference on Advanced Logistics and Transport*, Valenciennes, France, 2015.

[8] F. Calabrese, G. D. Lorenzo, L. Liu, and C. Ratti, "Estimating origin-destination flows using mobile phone location data," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 36–44, 2011.

[9] S. Isaacman, R. Becker, R. Cceres, S. Kobourov, M. Martonosi, J. Rowland, and A. Varshavsky, "Identifying important places in peoples lives from cellular network data," in *IEEE International Conference on Pervasive Computing*, San Francisco (CA), USA, 2011.

[10] M. Mamei, M. Colonna, and M. Galassi, "Automatic identification of relevant places from cellular network data," *Pervasive and Mobile Computing Journal*, 2016.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn, "Estimating origin-destination flows using mobile phone location data," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[12] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 993–1022, 2003.

[13] K. Farrahi and D. Gatica-Perez, "Discovering routines from large-scale human locations using probabilistic topic models," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 1, 2011.

[14] N. Eagle and A. Pentland, "Eigenbehaviors: Identifying structure in routine," *Behavioral Ecology and Sociobiology*, vol. 63, no. 7, pp. 1057–1066, 2009.

[15] G. Ferenstein. (2014) Uber shows new carpooling feature reduces traffic congestion 50% in pilot areas. [Online]. Available: http://venturebeat.com/2014/10/07

[16] L. Alexander, J. Toole, and G. M., "Asessing the impact of ride sharing on congestion using mobile phone data," in *Conference on the scientific analysis of mobile phone datasets (Netmob)*, Cambridge (MA), USA, 2015.

[17] N. Bicocchi and M. Mamei, "Investigating ride sharing opportunities through mobility data analysis," *Pervasive and Mobile Computing Journal*, vol. 14, pp. 83–94, 2014.

[18] B. Ciciy, A. Markopoulou, E. Fras-Martnez, and N. Laoutaris, "Quantifying the potential of ride-sharing using call description records," in *ACM International Workshop on Mobile Computing Systems and Applications*, Jekyll Island (GE) USA, 2013.

[19] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 111, no. 37, pp. 13 290–13 294, 2014.

[20] N. Bicocchi, M. Mamei, A. Sassi, and F. Zambonelli, "Opportunistic ride sharing via whereabouts analysis," in *IEEE International Conference on Intelligent Transportation Systems*, Canary Islands, Spain, 2015.

[21] R. Hall and A. Qureshi, "Dynamic ride-sharing: Theory and practice," *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 308–315, 1997.

[22] M. Brereton, P. Roe, M. Foth, J. Bunker, and L. Buys, "Designing participation in agile ridesharing with mobile social software," in *ACM OZCHI Conference of the Australian Computer-Human Interaction*, Melbourne, AU, 2009.

[23] I. Rahwan, S. Dsouza, A. Rutherford, V. Naroditskiy, J. McInerney, M. Venanzi, N. R. Jennings, and M. Cebrián, "Global manhunt pushes the limits of social mobilization," *IEEE Computer*, vol. 46, no. 4, pp. 68–75, 2013.

[24] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari, "Dynamic vehicle redistribution and online price incentives in shared mobility systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1567 – 1578, 2014.

[25] S. Yan, C. Chen, and Y. Lin, "A model with a heuristic algorithm for solving the long-term many-to-many car pooling problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1362 – 1373, 2011.

[26] K. Tan, K. Htet, and A. Narayanan, "Mitigation of vehicle distribution in an ev sharing scheme for last mile transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2631 – 2641, 2015.

[27] D. Pelzer, J. Xiao, D. Zehe, M. Lees, A. Knoll, and H. Aydt, "A partition-based match making algorithm for dynamic ridesharing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2587 – 2598, 2015.

[28] X. Chen, E. Santos-Neto, and M. Ripeanu, "Crowdsourcing for on-street smart parking," in *ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, Paphos, Cyprus, 2012.

Nicola Bicocchi is assistant professor in Computer Science at the University of Modena and Reggio Emilia, since December 2010. He received the PhD in Computer Science from the same University in 2009. He has been visiting researcher at University of Ulster and University of New Brunswick (Canada).

Marco Mamei is associate professor in Computer Science at the University of Modena and Reggio Emilia, since 2014. He received the PhD in Computer Science from the same University in 2004. He has been visiting researcher at Telecom Italia Lab (IT), Nokia Research Center (USA), Harvard University (USA), Cycorp Europe (SLO) and Yahoo Research (ES).

Andrea Sassi received the PhD at the Doctorate School of Industrial Innovation Engineering at the University of Modena and Reggio Emilia in March 2016. Now he is the lead developer of SmartHitch, a carpooling service for commuters and everyday travelers.

Franco Zambonelli is full professor of Computer Science at the University of Modena and Reggio Emilia. He got his PhD in Computer Science and Engineering from the University of Bologna in 1997. He has been scientific manager of the EU FP6 Project CASCADAS and coordinator of the EU FP7 Project SAPERE. He is ACM Distinguished Scientist, member of the Academia Europaea, and IEEE Fellow.