

Suggesting Just Enough (Un)Crowded Routes and Destinations

Claudia Cavallaro, Gabriella Verga, Emiliano Tramontana and Orazio Muscato

Department of Mathematics and Computer Science, University of Catania, Italy

Abstract

Though people like to visit popular places, for health-related concerns and due to the recent restrictions adopted around the world, gatherings should be avoided. When planning a trip, one has to consider both attractiveness in terms of general interest for the destinations, and the density of people gathering there. In this work, we propose a recommendation system aiming at offering users some suggestions on useful routes and destinations that balance both liveliness and overcrowding. Firstly, we use datasets storing GPS positions as a basis for the statistics on routes and destinations. Then, we use an accurate probability algorithm that estimates the number of people moving from one place to another in the city and accordingly we show a list of destinations to users. The destination points are filtered based on the user's preference on the density of people. A multi-agent system is used to handle the user requests to find a route for a trip, statistics on possible destinations, and suggestions to users. Thanks to our solution we can inform users on suitable routes and destinations, as well as alert them when a preferred destination is overcrowded.

Keywords

GPS trajectory, Recommendation systems, Movement predictions, Multi-agent system

1. Introduction

Currently, organising a trip should take into account the number of people that will gather in the chosen destination points, since it is necessary to avoid visiting a place that will become overcrowded to comply with the restrictions due to the Covid-19 influenza pandemic. Hence, an estimate of the number of people that will be in some place in a future time can be valuable for people moving and in situations where they could choose visiting some other place.

In previous works, the statistics accumulated over time are used to estimate a measure of traffic or gatherings [1, 2, 3, 4, 5]. Moreover, both popular online services, and other apps just count the number of people currently present in some place [6, 7, 8, 9]. However, statistics gathered in the past cannot be a reliable indication for the current situation that has to cope with e.g. restrictions on gatherings, lower capacity of public transport means, etc. due to the influenza pandemic. Additionally, a kind of real time measures of gatherings do not let other people plan their trip, hence understanding whether e.g. one hour later when arriving at the destination, the place will still be (un)crowded. A better estimate is therefore needed which

WOA 2020: Workshop "From Objects to Agents", September 14–16, 2020, Bologna, Italy

✉ claudia.cavallaro@unict.it (C. Cavallaro); gabriella.verga@unict.it (G. Verga); tramontana@dmi.unict.it (E. Tramontana); muscato@dmi.unict.it (O. Muscato)

ORCID 0000-0002-7169-659X (E. Tramontana)



© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

takes into account: (i) the current amount of people in some place, and (ii) the statistics on the number of people that being in some origin place typically flow to another place to visit later on. Moreover, an app behaving as an assistant agent is needed to timely inform interested people.

This paper proposes an approach to determine the probability that users are moving along some routes. Given the recordings of several user positions, we compute the probability for a user that being in some place A will move to another place B (i.e. a possible destination), hence when arriving in place B he will contribute to the number of people gathering there. By computing beforehand the probability that he will go to place B in a future time, we can guess whether a place will become overcrowded. Our proposed estimation of people destinations is based on the analysis of the co-occurrence of places statistically visited by an amount of people greater than a threshold. Moreover, we propose to give users an app that will let them know the amount of people that will gather to some areas that can be reached from the point where the user is. The app provides us support for suggesting the user possible destinations that are viable both in terms of distance, usefulness, and gathering. Moreover, the app provides means to collect data on the current amount of people in some place and then their trajectories. When collecting user data by means of the app we make sure that user privacy is preserved by providing only an approximate location to a central server.

Our approach can be useful in many contexts where estimating the number of people beforehand can be a crucial factor for a better service, such as e.g. when organising public transport, or for retailer, etc. Moreover, it could be enriched with data, coming from proper authorities, that reveal some places where a Covid-19-positive has been found. Then, by using our computed trajectories, we could give for other places the probabilities of having infection spreads.

The paper is organised as follows. Next section describes the related work. Section 3 explains our proposed solution. Section 4 illustrates the experiments and shows the viability of our approach. Finally, conclusions are drawn in Section 5.

2. Related work

This section offers an overview of the studies on multi-agent systems and the analysis of the movement of people.

Multi Agent System: a multi-agent system is a system with a significant number of independent agents that interact with each other [10]. In recent years, multi-agent systems have been widely used as being regarded suitable for systems with a modular architecture, thanks to their independence [10]. Generally, agents interact in three ways [11]: (i) each agent can communicate directly with any other agents (“autonomous agents”); (ii) agents communicate indirectly with each other via an intermediary (“facilitator”); (iii) all agents communicate with each other via an intermediary, however the agents can communicate with each other after the communication has been set up by the intermediary (“Mediator”). In the second case, the robustness can be poor and the overhead is relatively high but the intermediary acts as a protective wall for users privacy because agents do not communicate directly and it processes the information received from the users, decreasing their work [12]. On the other hand, the use of an intermediary has several advantages in terms of synchronization, reusability, scalability and modularity [13, 14].

Mobility monitoring: an accurate monitoring of user mobility provides support for efficient

resource usage. E.g., it could help avoid traffic congestion [1, 2, 3], give warnings or make targeted advertising by discovering the next place to visit [15], or simply study user behaviour [16]. The approach described in [4] identifies the routine behaviour of two sample of people by using a probabilistic approach (Latent Dirichlet Allocation) to extract 10 different positions shared by multiple users. Other approaches identify people movements. In [5], the proposed system observes GPS data in the urban area of Milan and creates an origin-destination matrix, then provides the similarity between two trajectories, leading to the discovery of mobility behaviours.

Moreover, to find typical trajectories of a group of people, the approach in [17] identifies flows using a grid with the Apriori algorithm [18], however noting only flows and not the frequently visited points of interests, nor the probability of some people moving to another place. In [19], the flows are identified by associating them with spatio-temporal trajectories shared by multiple users heading in the same direction. Finally, other approaches use machine learning techniques, e.g. the approach in [20] is based on a clustering algorithm, and uses an unsupervised learning solution for an automatic lane detection in multidirectional pedestrian flows.

Our proposed work uses a multi-agent system communicating with a server that acts as an intermediary and predicts movements by means of an innovative and reliable mathematical solution. Unlike the work presented in [4], our goal is to detect the foreseeable routes by computing their probability, instead their method determines the probability that a group of users is moving together. The work presented in [20] uses an unsupervised learning approach and aggregates instantaneous information on the position and speed of pedestrians to form clusters, calculated on short time windows. We use a fixed grid to group people positions into cells, and then compute the probabilities using the data arriving from the agents in real time. The paper [21] presents personalised recommendations for guiding tourists through the city of Melbourne by observing their actions. This system is modelled as a Markov decision process that recommends the user in sequence the next place to visit. However, unlike the StayPoint analysis presented here, it does not consider the stationary nature of visitors over a period of time and this is a key element in avoiding overcrowding. In [17] the frequent *corridors*, i.e. routes, were found on a grid through Apriori algorithm [18], while in this work we initially find the areas having highly visited points of interest, hence giving great importance to the user stay time. Moreover, we consider the Confidence and Lift metrics used in the Market Basket Analysis to know the probability of displacement and therefore predict the contagion areas that are continuously updated.

3. Proposed Approach

We aim at determining the probability that users are moving from one point to another point. Such a probability is then used to provide recommendations accordingly. Recommendations, alerts, or user requests, are communicated by means of a smartphone app. Therefore, our proposed solution comprises two parts: (i) an algorithm that determines the probability of people movements, and (ii) an app on the users device to track movements and suggest destinations.

3.1. Determining Probability of Movement

To determine the probability of movement we perform two steps. The first step consists in determining shared people flows from the points recorded during the previous movements of each user. Then, the second step consists in obtaining statistics on the amount of people that being in point *A* subsequently go to point *B*.

By analysing every *GPS trajectory*, i.e. the set of recorded GPS points temporally ordered, we extracted its *StayPoints (SPs)*. They are the centres of the areas within which a user stays for more than a certain time: for some reason that area is of interest. Then, the geographical area where the *SPs* of all dataset are located has been discretised by means of a grid, made up of equal *Square Cells*. Each determined *SP* has been associated with a single square cell if it is contained in that space. Sure, a cell could contain multiple *SPs* if these are close enough, depending on the width of the cell. Cells that did not contain any *SP* have not been considered.

We then determined the subset of *frequently visited cells* consisting of all the cells that having at least one *SP* within them have been visited at least by 10% of the people. For the sake of reliability, we compute only the statistics between the frequently visited cells, and we consider the Confidence as used by the Market Basket Analysis. Confidence denotes the percentage of trajectories frequently visiting a cell *B* which also frequently visit cell *A*. I.e. for a value of Confidence higher than a threshold (set as 60% in our experiments), we can assert that a large group of people having visited cell *A* moves together to cell *B*. Confidence is an estimation of conditioned probability. Two or more cells for which there is a Confidence higher than 60% that have been visited by a large group of people are dubbed *co-visited cells*.

Then, we check the reliability of the association rules obtained ($A \Rightarrow B$) through *Lift*, which will confirm that the transition of a user from the *SP* in *A* to the *SP* in *B* has a positive correlation.

3.2. A Multi-Agent Recommendation System

In general, an agent, according to Wooldridge [22], is merely "*a software (or hardware) entity that is situated in an environment and is able to autonomously react to changes in that environment*". Each agent has the basis to learn and communicate, and in our case, learning takes place by capturing the user GPS positions and, communication is realised by connecting to a centralised server, which alerts all agents when needed and stores the geographical coordinates of the points visited by users. Figure 1 shows the main components of the proposed multi-agent system.

An agent runs on a smartphone as an app in order to receive suggestions on possible destination. The agent offers recommendations highlighting any 'warm', that is very crowded, or 'cold', that is uncrowded, place, using the statistics gathered as described in the previous section. For this, the agent periodically reads the user position and checks whether a known StayPoint (*SP*) is nearby. Then, the agent communicates to the server whether it is close to a *SP*. This lets the agents contribute in determining the number of people close to a *SP*, rather than giving their actual position, hence preserving the user's privacy. In this context, the privacy protection is intended to prevent the disclosure of information relating to the exact location of the user. Figure 2 shows the app providing information to the user.

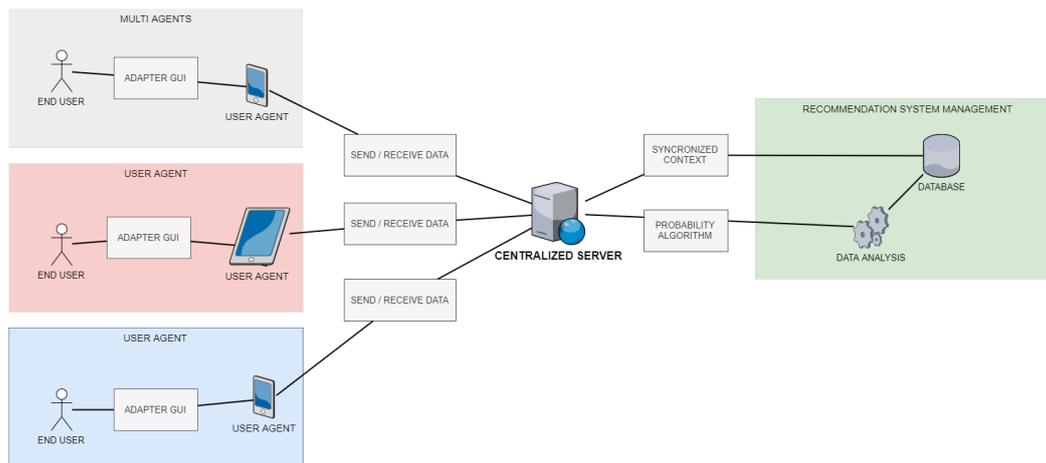


Figure 1: System architecture showing the interaction between agents and server: each agent sends his preference for crowded places and where he is, the server gathers data and creates recommendations.

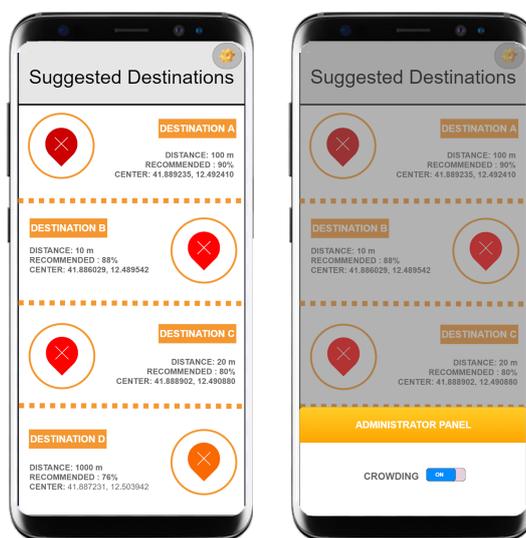


Figure 2: User communicates with agents via application GUI. The left panel shows the list of destinations suggested by the multi-agent system and the right panel is the administration view where the user gives his preferences on (un)crowded places. The colour of the icons represents the intensity of crowding, that is, more (less) red equals more (less) crowded.

The server, having acquired by the user the position of the nearest *SP*, returns the list of other *SP*s that could be visited according to the probability estimate of passing through that point (0 equals low probability, 1 equals high probability). In this way, we create a Collaborative Filtering based recommendation system [23], as it is based on the choices of other users.

Finally, the user, through an administration panel, can set with a flag, if he prefers 'warm' or 'cold' places. Thus, the agent, based on the users choice and the list received from the server, determines information to show and then suggest. Destinations are displayed as a map or a

list. All agents are independent of each other and since they extrapolate data directly from the device they are reliable, making the architecture stable and trustworthy.

4. Experiments

This section describes the experiments carried out using a dataset that collects real movements from one part of the city to another by taxis and/or people. Positions have been gathered by periodically reading geo-coordinates from tablets or smartphones. The experiments focus on data analysis for determining the probabilities of moving from one StayPoint to another as described in Section 3. This approach is used in our centralized server in order to select the list of suggestions to send to the agents. The used dataset allows us to simulate the behaviour of a reasonable number of users, showing the usefulness of the app. Over time, data are updated as provided by agents. Below we describe the dataset used in our experiments, then the tests carried out, and the results that have been found.

4.1. Dataset

The dataset used to perform our tests is *Cabspottingdata* [24] and includes the trajectories collected in May 2008 by 536 taxis, for a total of 11, 219, 424 GPS points. Cab mobility traces are provided by the Exploratorium—the museum of science, art and human perception through the cabspotting project¹. To gather data each vehicle was outfitted with a GPS tracking device that was used by dispatchers to efficiently reach customers. Data were sent from each cab to a central receiving station, and then delivered in real-time to dispatch computers via a central server. Each mobility trace file, associated to a taxi ID, contains in each line: latitude, longitude, occupation, timestamp. Where latitude and longitude are in decimal degrees, the occupation indicates whether a taxi has a fare (1 = busy, 0 = free) and the time is in the UNIX era format. The area covered by these routes corresponds to the county of San Francisco of USA and its surroundings in California, with maximum and minimum longitude and latitude = $[-127.08143; 32.8697] \times [-115.56218; 50.30546]$. The total size of the trajectories registered with a customer on the taxi consists of 5, 017, 659 points.

4.2. Tests carried out to find flows and StayPoints

A trajectory T is an ordered sequence of GPS points, in which the positions occupied (for example by a vehicle) and the timestamps associated with them are recorded chronologically. Each position is represented by latitude and longitude of the geographical point. $T = \{p_0 = (x_0, y_0, t_0), p_1 = (x_1, y_1, t_1), \dots, p_n = (x_n, y_n, t_n)\}$, where $\forall i \in [0, n]$, $p_i = (x_i, y_i, t_i)$ with $t_i < t_{i+1}$, and x_i , y_i and t_i represent longitude, latitude, and timestamp, respectively.

The first step was the data cleaning in order to eliminate noise, due for example to GPS errors. It was performed by computing the instantaneous speed of each point of the rides recorded on the taxi. The maximum acceptable speed threshold has been set for $150 \frac{km}{h}$. The trajectories have been summarised for the comparison of the distance, considering for each

¹<http://cabspotting.org>

path two successive points in temporal order only if they were at a minimum distance of 140 *m*. This was done to decrease the size of the dataset and therefore will allow a reduction in the execution time of the algorithm.

To carry out a statistical analysis, 90% of trajectories were randomly selected, and this set was the *Train* set for the the flow detection algorithm. The complementary set, that is the remaining 10% of the trajectories, consists in the *Test* set, that is the verification set. We considered 6 time slots of 4 hours each, to visualise the movement of the vehicles at different times and the trajectories were therefore split according to the 6 time slots. To identify the sub-trajectories common to different users in the same time slot, a maximum tolerance distance was set between two different points of different users as 280 *m*. The distance between two points was computed by using the *Haversine distance*, which given two points $P_i(lt_i, lg_i)$ and $P_j(lt_j, lg_j)$ characterised by latitude and longitude in decimal degrees returns their distance in meters considering the curvature of the earth:

$$d(P_i, P_j) = 2R \arcsin \sqrt{\sin^2 \frac{lt_i - lt_j}{2} + \cos lt_i \cos lt_j \sin^2 \frac{lg_i - lg_j}{2}}$$

where R is the mean radius of the earth.

We define *flows* as close sub-trajectories, belonging to different users, spatially similar and recorded in the same time slot. The density of a flow is the number of users that pass through it. For detecting flows in this dataset, the minimum density threshold was set to 25. According to these parameters, 12 flows were identified, ranging from 1 to 2 km in length. The minimum density of the flows found is 26 taxis, while the maximum density found is 192 taxis. Then, by taking the complement of the trajectory sample (10% of the taxis, as the test set) we checked where their GPS points were compared to the previous train set. We found that the points of the test set intersect with the 12 paths identified on the train set. Another check was carried out by confirming the correspondence of the points of the flows on a map. It consists of the process of matching the coordinates of the obtained flows and the road segments, and assessing that there are no external points with respect to road segments (see Figure 3).

We apply the StayPoint detection algorithm to each trajectory (more details can be found in [25]) with time threshold, *TimeThr*, equal to 10 minutes, and distance threshold, *DistThr*, 100 meters. Such thresholds should suffice to select the positions in which a user dwells (in several *SPs*) as he finds the place interesting, and removes the locations where a user is stopping because e.g. he is blocked at the traffic light.

The execution time for the StayPoints detection algorithm on the whole Cabspotting dataset (536 taxis and more than 11*M* points) was 36 minutes and 54 *s*. We obtained a total of 4261 *SPs*, which is an average of 8 *SPs* per vehicle journey. The results show that 98% of users have at least one StayPoint associated with their trip (523 users out of 536). The implementation of StayPoints detection algorithm used Python and the experiments were executed in a host having an Intel Xeon CPU E5-2620 v3 2.40GHz, with RAM 32 GB.

Figure 4 shows the recorded trace for each trip in blue, and the detected *SPs* in yellow. Figure 5 shows the detected flows in magenta and the *SPs* in that area in green.

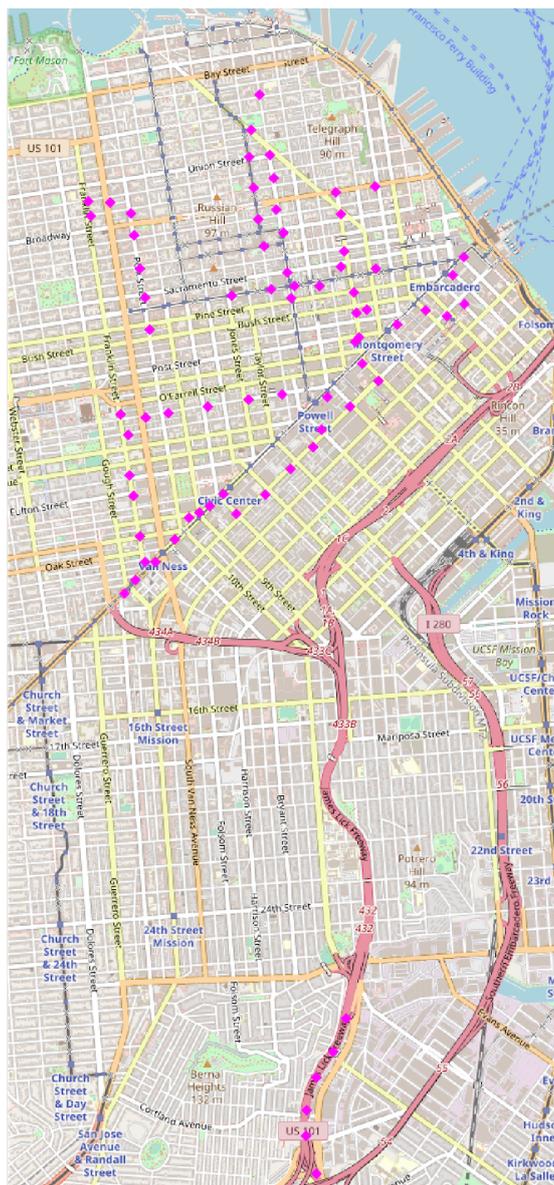


Figure 3: Flows detected for the Cabspotting dataset.

4.3. Movement prediction

For predicting the movements of people, firstly a grid was built, which covers the map, made up of square cells with a side of 1 km. Such a grid lets us discretise the data and estimate the probability of movement from a cell having some *SPs* inside it to another cell also having at least one *SP*. Two distinct geographical areas comprising some *SPs* are represented as two square cells without intersection, therefore a space partition is formed. Figure 6 shows such a grid, having size 80×46 cells (latitude by longitude), and the obtained *SPs* are mapped in to the grid

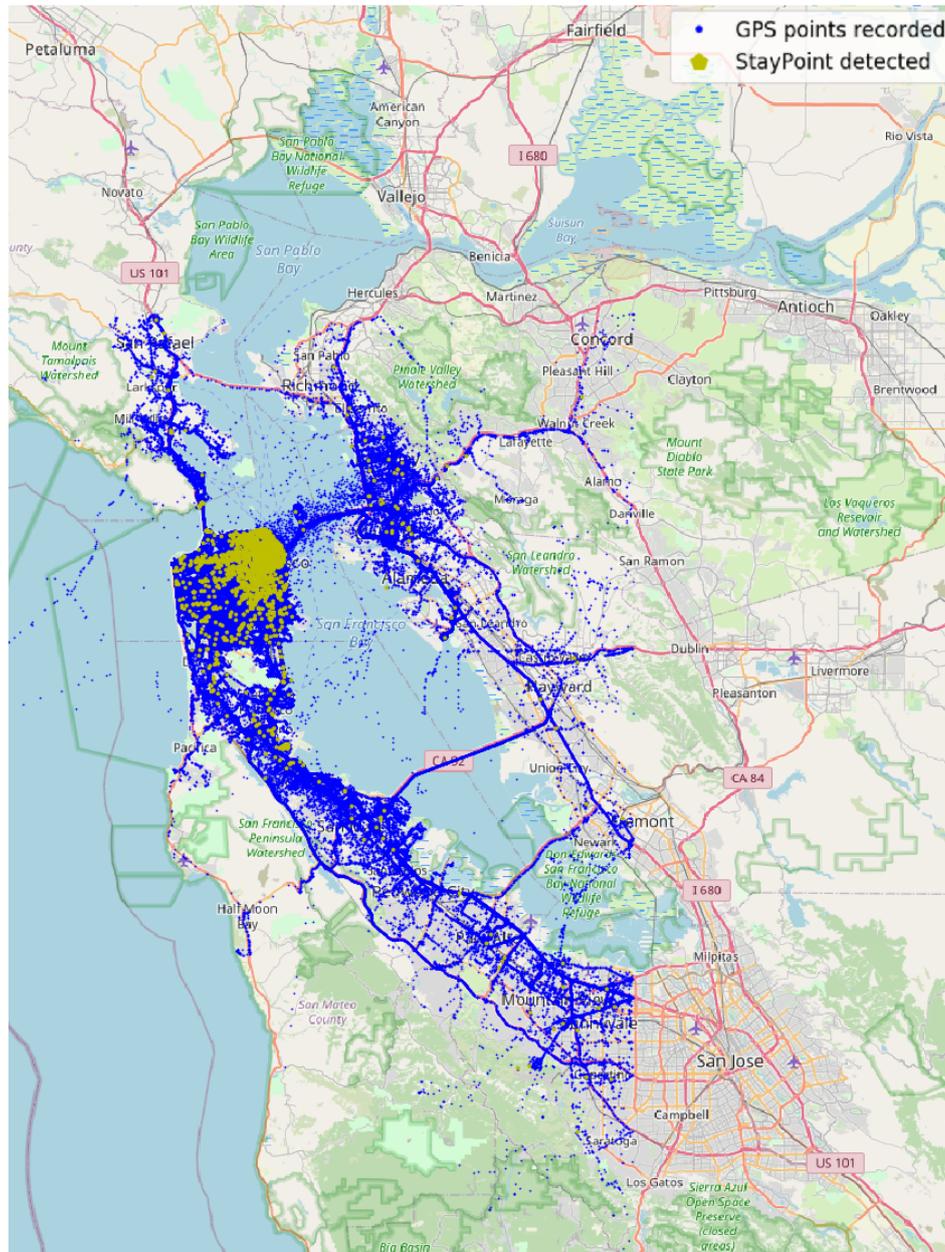


Figure 4: Blue points for trajectories and StayPoints obtained in yellow.

and shown as red dots. Some areas consisting of nearby cells have many more *SPs* than others, hence red dots are more dense in some areas than others, as shown in the said figure.

In order to determine whether a cell *A* is a frequent destination, the *Support* for each cell was calculated. The *Support* is the ratio between the number of trajectories that contain the cell and the total number of trajectories. If this ratio exceeds a certain threshold, i.e. if cell *A*

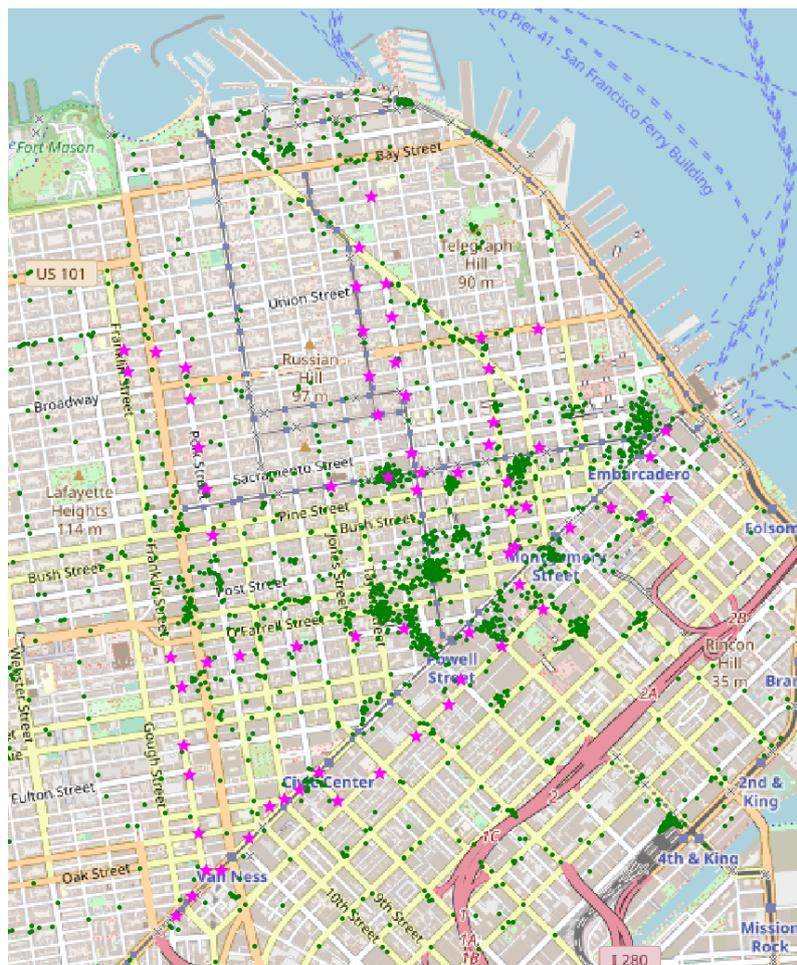


Figure 5: A zoomed in map of an area in Figure 4, showing flows in magenta and the nearby StayPoints in green.

is crossed by a certain number of different trajectories (10% value was chosen for Minimum Support, i.e. 0.1), then the cell (containing one or more *SPs*) will be a *frequently visited cell*.

Our experiments on the above said taxi dataset have shown that there are 43 cells visited by a number of users greater than or equal to 52. I.e. we can say that in the dataset there are 43 frequently visited *SPs* cells. This means that there has been a probable meeting in that cell, as users have remained stationary in the same time slot in the same cell. Data are updated in real time through the agents running on smartphones as an app, therefore the Minimum Support is fixed, however the number of frequent cells in output and the position of these frequently visited cells will vary over time.

By lowering the Minimum Support, i.e. the threshold of the minimum amount of people sharing the same cell, the number of cells considered as having a sufficient amount of people will increase and then the number of cells considered overcrowded will increase. In order to compute association rules only between the frequently visited cells in the dataset, we considered

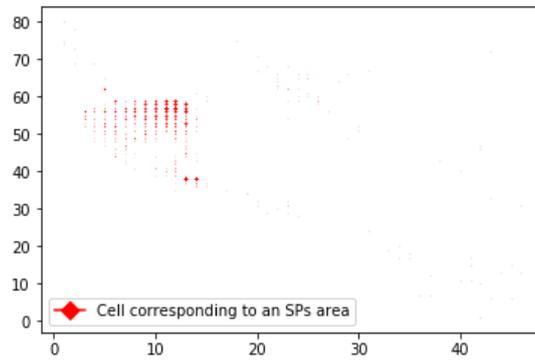


Figure 6: Grid formed by squared cells of 1 km per side, each red dot represent a cell having at least one SP.

the *Confidence* of the Market Basket Analysis for our approach.

Given two cells called A and B we have that:

$$Support(A \Rightarrow B) = \frac{Frequency(A, B)}{N}$$

the Support of the association rule ($A \Rightarrow B$) denotes the percentage of trajectories containing A which contain also B , where N is the total number of trajectories.

$$Confidence(A \Rightarrow B) = \frac{Support(A \Rightarrow B)}{Support(A)}$$

Hence, confidence is an estimation of conditioned probability, which can be expressed as follows:

$$Confidence(A \Rightarrow B) = \frac{p(A \cap B)}{p(A)} = P(B|A).$$

In Market Basket Analysis, *Confidence* is the probability of purchasing item B , said *consequent*, given the purchase of object A , said *antecedent*, within the same transaction. The higher the *Confidence*, the greater the reliability of the ($A \Rightarrow B$) rule (more details can be found in [26]). In our context, the value computed as the $Confidence(A \Rightarrow B)$ gives the probability that a user is in a *SP* in cell B moving there together with at least 10% of the total number of users, if he has already been in cell A and dwelling in one of its *SPs*.

Going forward along this procedure, we compute $Confidence(A, B \Rightarrow C)$ and after that $Confidence(A, B, C \Rightarrow D)$, in order to determine a common path that crosses several cells having highly visited *SPs*. We compute

$$Confidence(A, B \Rightarrow C) = \frac{Frequency(A, B, C)}{Frequency(A, B)}$$

and so on.

Therefore, the results obtained are useful to predict the number of gatherings on some place. Moreover, given that there is knowledge about an infected person on some area, our results can

be used to predict whether a user can be potentially infected (as his trajectory is estimated), and predict who else he will infect (i.e. people whose trajectories are expected to pass through the same areas).

The Confidence limit is due to the fact that it does not consider the Support of the item on the right side of the rule and therefore does not provide a correct evaluation in case the groups of items are not stochastically independent.

A measure that takes this eventuality into account is $Lift(A \Rightarrow B)$, defined as:

$$Lift(A \Rightarrow B) = \frac{Confidence(A \Rightarrow B)}{Support(B)} = \frac{p(A \cap B)}{p(A) * p(B)}$$

$Lift(A \Rightarrow B)$ takes into account the importance (the Frequency) of B . Using such an amount, then we can say

- if $Lift > 1$ the events are positively correlated;
- if $Lift \leq 1$ the events are negatively correlated or independent.

Therefore, Lift indicates how the occurrence of one event raises the occurrences of the other. At this point, the setting would be a Minimum Confidence (0.6 i.e. 60%) to skim the results and obtain only the association rules that had a higher Confidence and also a Support higher than the Minimum Support (0.1 chosen), such a setting is named *Strong rules*. Finally, these rules were checked with the Lift, the last column of Table 1. Then, for the Association Rule ($[2587] \Rightarrow [2588]$) in row 6 the events of movement from cell A to cell B are negatively correlated.

The left panel in Figure 7 shows the plot of every Strong Rule obtained as a point, as a value for its Support and Confidence (the latter according to the Support). For association rules with higher support the Confidence, that is the probability of moving to the frequently visited cell B , decreases. The Lift and the Confidence of the Strong Rules obtained are directly proportional, as we can see in the right panel of Figure 7. The Pearson correlation coefficient between them is 0.9999999999999999 and this implies an exact linear relationship.

Moreover, the results tell us that the probability of transitioning from one cell with SPs to another is high even in correspondence with the indicated flows and that different highly visited cells having SPs belong to different flows. For each cell we have checked which taxis passed there and which passed at a later time on other flows passing through other frequently visited cells.

5. Conclusions

Having an educated guess on the amount of people that will gather in some place before planning a trip can be very useful to avoid overcrowded places and to keep with the current regulations. We have proposed an approach for predicting the probability of people moving to some destinations when it is known that a certain amount of people is in some other place. We use an app that senses the position of people and sends to a server such data. Then, such an amount is useful, together with previous statistics, to estimating the amount of people in

Table 1

A set of Cells and the related Support (Sup), Confidence (Conf), and Lift, when cell B is 2588, $Support(B) = 0.671128$, and having Strong Association Rules: Minimum Confidence 60% , Minimum Support 10%

$(A \Rightarrow B)$	A	$Sup(A)$	$Sup(A \Rightarrow B)$	$Conf(A \Rightarrow B)$	$Lift(A \Rightarrow B)$
1	[1715]	0.395793	0.284895	0.719807	1.072533
2	[1716]	0.313576	0.223709	0.713415	1.063008
3	[2405]	0.151052	0.108987	0.721519	1.075084
4	[2541]	0.137667	0.099426	0.722222	1.076132
5	[2543]	0.242830	0.175908	0.724409	1.079391
6	[2587]	0.435946	0.292543	0.671053	0.999888
7	[2633]	0.145315	0.103250	0.710526	1.058704
8	[2634]	0.281071	0.206501	0.734694	1.094715
9	[2635]	0.235182	0.177820	0.756098	1.126607
10	[2679]	0.202677	0.147228	0.726415	1.082379
11	[2680]	0.242830	0.175908	0.724409	1.079391
12	[1715, 1716]	0.149140	0.112811	0.756410	1.127073
13	[2587, 1715]	0.156788	0.112811	0.719512	1.072094
14	[2634, 2587]	0.141491	0.101338	0.716216	1.067183

another place at a later time. The experiments that we have performed on previously gathered geographical locations have shown the viability and reliability of our approach.

The more people use the app the more the approach would give a correct estimate. To make the approach more robust, it could be extended in order to include data available online from other services that give indications on queues, road traffic, gatherings.

Future work will consider the geometry of stations, museums, etc. of some popular destinations to compute the average distance of people given the estimated size of crowds. Moreover, how alerts are spread will consider both the people already in some place and the people moving towards it.

Acknowledgments

The authors acknowledge the support provided by means of a PO FSE 2014-2020 grant founded by Regione Siciliana, and by project TEAMS-TEchniques to support the Analysis of big data in Medicine, energy and Structures–Piano di incentivi per la ricerca di Ateneo 2020/2022.

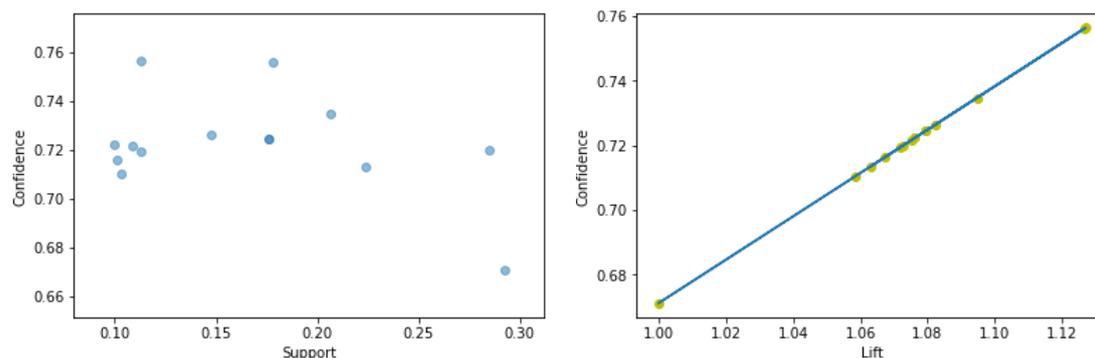


Figure 7: The left panel shows the Support vs Confidence for Strong Rules obtained by our analysis and given as rows in Table 1; the points show a pair of cells (A and B), or a triple of cells (for the last three rows of Table 1). The right panel shows Lift vs Confidence in this test, as for the points shown in the left panel.

References

- [1] P. Castro, D. Zhang, S. Li, Urban traffic modelling and prediction using large scale taxi gps traces, in: Proceedings of of International Conference on Pervasive Computing, 2012, pp. 57–72. doi:10.1007/978-3-642-31205-2_4.
- [2] J. Lee, J. Han, X. Li, A unifying framework of mining trajectory patterns of various temporal tightness, IEEE Transactions on Knowledge and Data Engineering 27 (2015) 1478–1490.
- [3] Z. Wang, M. Lu, X. Yuan, J. Zhang, H. van de Wetering, Visual traffic jam analysis based on trajectory data, IEEE Transactions on Visualization and Computer Graphics 19 (2013) 2159–2168.
- [4] N. Bicocchi, M. Mamei, Investigating ride sharing opportunities through mobility data analysis, Pervasive Mobile Computing 14 (2014) 83–94.
- [5] R. Trasarti, et al., Exploring real mobility data with m-atlas, in: Proceedings of Machine Learning and Knowledge Discovery in Databases, Springer, 2010, pp. 624–627.
- [6] J. Cranshaw, R. Schwartz, J. Hong, N. Sadeh, The livelihoods project: Utilizing social media to understand the dynamics of a city, in: Proceedings of AAAI Conference on Weblogs and Social Media, 2012.
- [7] C. Berzi, A. Gorrini, G. Vizzari, Mining the social media data for a bottom-up evaluation of walkability, in: Proceedings of International Conference on Traffic and Granular Flow, Springer, 2017, pp. 167–175.
- [8] B. P. L. Lau, M. S. Hasala, V. S. Kadaba, B. Thirunavukarasu, C. Yuen, B. Yuen, R. Nayak, Extracting point of interest and classifying environment for low sampling crowd sensing smartphone sensor data, in: Proceedings of IEEE Pervasive Computing and Communications, 2017.
- [9] C. Cavallaro, G. Verga, E. Tramontana, O. Muscato, Eliciting cities points of interest from people movements and suggesting effective itineraries, Intelligenza Artificiale (2020). doi:10.3233/IA-190040.

- [10] K. P. Sycara, Multiagent systems, *AI Magazine* 19 (1998) 79. URL: <https://www.aaai.org/ojs/index.php/aimagazine/article/view/1370>. doi:10.1609/aimag.v19i2.1370.
- [11] W. Shen, D. Norrie, Facilitators, mediators or autonomous agents, in: *Proceedings of International Workshop on CSCW in Design, 1997*, pp. 119–124.
- [12] Q. Liu, L. Gao, P. Lou, Resource management based on multi-agent technology for cloud manufacturing, in: *Proceedings of Electronics, Communications and Control (ICECC), IEEE, 2011*, pp. 2821–2824.
- [13] J. Z. Hernández, S. Ossowski, A. Garcia-Serrano, Multiagent architectures for intelligent traffic management systems, *Transportation Research Part C: Emerging Tech.* 10 (2002) 473–506.
- [14] E. Tramontana, Minimising changes when refactoring applications to run multiple threads, in: *Proceedings of IEEE Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 2018*, pp. 713–714.
- [15] G. Verga, A. Fornaia, S. Calcagno, E. Tramontana, Yet another way to unknowingly gather people coordinates and its countermeasures, in: *Proceedings of Internet and Distributed Computing Systems (IDCS), Springer LNCS 11874, Naples, Italy, 2019*, pp. 130–139.
- [16] A. Noulas, S. Scellato, N. Lathia, C. Mascolo, Mining user mobility features for next place prediction in location-based services, in: *Proceedings of IEEE International Conference on Data Mining, 2012*, pp. 1038–1043.
- [17] C. Cavallaro, J. Vitrià, Corridor detection from large gps trajectories datasets, *Applied Sciences* 10 (2020) 5003. doi:10.3390/app10145003.
- [18] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: *Proceedings of International Conference on Very Large Data Bases, Morgan Kaufmann, Los Altos, CA, 1994*, pp. 478–499.
- [19] E. Tramontana, G. Verga, Demo: Get spatio-temporal flows from gps data, in: *Proceedings of IEEE International Conference on Smart Computing (SMARTCOMP), Taormina, Italy, 2018*, pp. 282–284.
- [20] L. Crociani, G. Vizzari, A. Gorrini, S. Bandini, Identification and Characterization of Lanes in Pedestrian Flows Through a Clustering Approach, volume 11298, Springer Verlag, 2018, pp. 71–82. doi:10.1007/978-3-030-03840-3_6.
- [21] F. de Nijs, G. Theodorou, N. Vlassis, M. M. de Weerd, M. T. J. Spaan, Capacity-aware sequential recommendations, in: *Proceedings of International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2018*, p. 416–424.
- [22] M. Wooldridge, N. R. Jennings, Intelligent agents and multi-agent systems I, *Applied artificial intelligence* 9 (1995).
- [23] P. Melville, V. Sindhwani, Recommender systems., *Encyclopedia of machine learning* 1 (2010) 829–838.
- [24] M. Piorkowski, N. Sarafijanovic-Djukic, M. Grossglauser, CRAWDAD dataset epfl/mobility (v. 2009-02-24), 2009. doi:10.15783/C7J010.
- [25] C. Cavallaro, G. Verga, E. Tramontana, O. Muscato, Multi-agent architecture for point of interest detection and recommendation., in: *Proceedings of Workshop from Objects to Agents (WOA), Parma, Italy, 2019*, pp. 98–104.
- [26] J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.