User Assistance for Predicting the Availability of Bikes at Bike Stations

Claudia Cavallaro[1], Emiliano Tramontana[2]

[1]INFN-CNAF Centro Nazionale per la Ricerca e lo Sviluppo nelle Tecnologie Informatiche e Telematiche, Viale Berti Pichat 6/2, 40127 Bologna, Italy.

[2]Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria, 6, 95125 Catania, Italy.

Abstract

Generally, assistant agents have been employed to recommend users some goods or services they could be interested in. Moreover, thanks to devices recording user geographical positions, recommendation systems have been developed to propose places to visit. This paper proposes an approach for making an estimate of bikes available at bike stations, hence facilitating the use of such a transport means. I.e. users made aware of bike availability beforehand can choose such a transport mode more easily. By using a fast algorithm analysing data recording all bike movements, we obtain an accurate estimate of where bikes will be in the near future. This is possible by determining the frequent paths of bikes, according to their starting points, and the likely destinations. Thanks to such estimates, users can be alerted beforehand about the desired bike at their preferred bike station. Then, a message will be sent when the bike is in the station. Assistant agents giving such alerts are provided to users as an app on their smartphone.

Keywords

Assistant agents, Recommendation system, Bike-sharing, Forecast, People gathering

1. Introduction

Plenty of data are available from, and are regularly produced by, personal devices tracking the movement of people in terms of GPS coordinates. Such data gathered from devices, as e.g. smartphones, smart watches, cars, are usually sent to an aggregator host and then used to gain knowledge on the the behaviour of people, on possible routes, on the density of people in some places, etc. [1, 2, 3]. E.g. it is possible to determine the most visited tourist places in a city and the usual paths traversed to reach such places [4]. Apps using data gathered from the movement of people have been offered to users to suggest lively places to visit, e.g. in a certain time-frame [5, 6], paths inducing some kinds of emotions [7, 8], etc. An app on a smartphone, while contributing to data gathering, provides users with a means to be notified of potential useful information, such as e.g. a nearby point of interest. By giving users suggestions, and acting on behalf of the user for finding the best spot, by communicating with a server, such an app can be seen as an *agent* assisting the user in her daily routine.

WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy
claudia.cavallaro@cnaf.infn.it (C. Cavallaro); tramontana@dmi.unict.it (E. Tramontana)
0000-0003-3938-0947 (C. Cavallaro); 0000-0002-7169-659X (E. Tramontana)

In previous studies, the analysis of GPS coordinates has been performed by using several techniques, such as firstly computing the point of interests, and then determining how frequently each user can be found nearby. Apriori algorithm has been used to find how frequently a set of places have been visited by users passing through the same path [6]. However, the execution of the Apriori algorithm takes a long time [9]. Apriori has a complexity of $O(2^n)$, where n is the number of places to be considered. Hence, such a complexity requires a great deal of computation.

Sure, for analysing the ever growing amount of GPS coordinates recorded, it is paramount to have a solution that minimizes computation time. In this work, we consider the position of bikes in a bike sharing scenario and compute the common paths and the probable destinations with a certain probability. This is useful to give users that wish to rent a bike the availability of bikes at bike stations in the near future. Moreover, should restrictions on the gathering of people in public spaces persist (due to the pandemic outbreak), thanks to the available estimate on the number of bikes arriving at a bike station, the devised app can be handy to give people an alert beforehand suggesting the best time for approaching the bike station.

This paper proposes a novel approach to process data concerning the geographic positions of bikes, to find common paths and infer possible destinations. The used algorithm is FP-Growth that has been shown as having a lower execution time, when compared with Apriori [10, 9], as FP-Growth is a linear time algorithm[1]. This approach is novel as FP-Growth has never been applied for the said objective before.

The rest of the paper has the following structure. Section 2 gives the comparison with the relevant related work. Section 3 describes the proposed software architecture to assist users in finding bike stations with available bikes. Section 4 introduces the dataset used for experiments. Section 5 details how the FP-Growth algorithm has been used for geo-spatial data. Section 6 shows the results when analysing a big amount of data related to bike positions withe the proposed approach. Finally, Section 7 reports our concluding remarks.

2. Related works

The Spatio-Temporal Data Mining problem has been addressed in many works in the literature by using different techniques. This section presents some of the most relevant approaches having similarities with respect to our approach.

In [11], Pensa et al. addressed the problem of finding frequent sequential patterns for a dataset of real vehicle GPS trajectories tracked in Milan, Italy. The proposed algorithm transforms spatio-temporal trajectories into sequences of regions of interest (ROI), based on a discretization of the working space through a regular grid. The authors measured the similarity of each pair of patterns as support density and length. By using the technique PrefixSpan [12], ROI's sequences were represented as a tree, where each node is a ROI. Then, each branch of the tree was associated with the support in each trajectory and subtrees having infrequent ROIs were removed. Finally, they presented the framework P2kA that anonymizes the dataset and thus preserves the privacy of users in the frequent paths extracted. Unlike our proposal, in [11] the objective was not aiming at making predictions on future destinations, while our strategy

[1] https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/

allows us, with a certain probability, to identify the next bike station reached by the same group of cyclists racing together. Moreover, our work need not obscure certain information on the data, because the ID associated with a registered route refers to a specific rented bike in a time slot and not to the customer's card.

Other data mining methods, such as DBSCAN [13], were used for the purpose of identifying parts of common routes or shared Points Of Interest [4]. In [14], Crociani et al. offered an unsupervised learning approach for an automatic lane detection in multidirectional pedestrian flows, taking into account the angular distance during the movement.

Regarding recommendation systems, in [7], of Quercia et al. gathered metadata from the pictures in Flickr, and determined routes to be suggested according to parameters of beauty, quietness and happiness. They computed the probability that an individual visits a certain destination because it is pleasant. In the literature other algorithms, such as the one presented in [15], recommended public transport routes that include both short walks and reduced waiting times. Our forecast indicates how many cyclists of a given group will subsequently head to a certain point, according to previous traffic data. This aims to: (i) show bikes availability, and (ii) alert users in case some places will be overcrowded. Another travel recommendation system that analyzes GPS trajectories was presented in [16]: the authors used HITS [17] algorithm, and recommendations were based on the travel experiences of the users (hub scores) and the interests of the road segments (authority scores).

In [18], Cecaj and Mamei presented a useful way to extract associative rules, with the Eclat algorithm, to find co-locations of companies in industrial agglomerations. Our approach mines the frequent stations from bike trips and differs for the purpose of forecasting future displacements. Some differences between three frequent mining pattern algorithms (Apriori, Eclat and FP-Growth) are shown extensively in [19].

3. Multi-Agent Architecture

Our proposed system comprises an app running on a smartphone and acting as an assistant *agent* for the user. Such an app communicates with a server side both to send gathered data, i.e. geographical coordinates, as well as user requests. Moreover, the app receives useful data that the server has gained from data analysis and selected as possibly useful to the user according to her preferences and requests.

In our application, the user is interested in finding a given type of bike available in one among a few nearby bike stations. Hence, according to our approach, the server performs data analysis to estimate the future availability of bikes and sends to interested users, on their app, an alert when the bike is likely to be available and when. The interested user, by means of the app can express the preference to lock the bike, then and the server will give her confirmation of availability. Figure 1 shows a sketch of some dialog panels a user can be provided with: from the left there is an opening logo, then a screen for letting the user input her preferences on the bike she wishes to rent, then a screen for the preferred bike stations, and finally a screen showing the number available bikes at the selected bike stations. The user preferences are sent from the mobile app (assistant agent) to a server, whereas the alert and details on bikes available are received from the server on the mobile app.

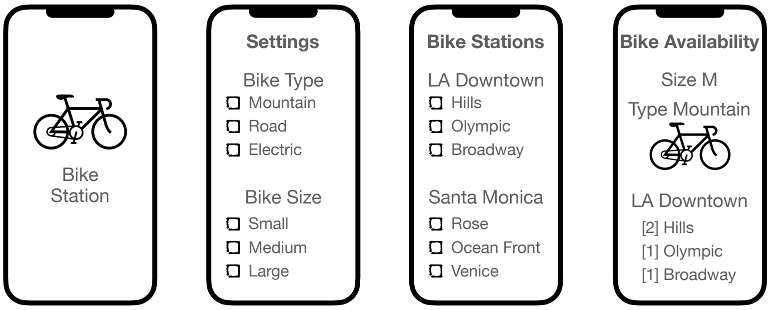

Figure 1: User interface provided by the app on the smartphone by the assistant agent.

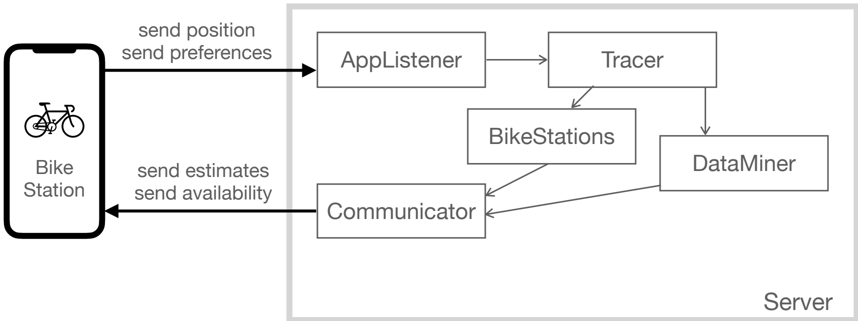

Figure 2: Software architecture for the multi-agent system.

Agents, as apps on a smartphone, communicate indirectly to each other, i.e. they rely on the server to send data, as their location or trajectory when having rented a bike, since such data are useful to make forecast of future destinations, hence give estimate of bike availability to others. The indirect communication between agents ensures the privacy of users, as a user location and identity cannot be discovered by other users. The server collects the ID of the bikes and the position, not the identity of the user, hence the safety of users is enhanced. Moreover, a user can set to have her coordinates being altered by a random value within a range to make it uncertain the exact position even to the server further enhancing the privacy of the user [20].

Figure 2 gives an overview of the software architecture. The app on the smartphone communicates with a server side, which is a host where a component called *AppListener* handles incoming data transmitted by each app (therefore by each user). The data are: the bike positions sent periodically, and user preferences, i.e. the type of bike, the station of interest, rental time, etc. The *AppListener* component sends the position data to the *Tracer*, and the latter updates the *DataMiner* with new data periodically. The data on the stations and the bikes present in the stations are kept by *BikeStations*, and updated according to the positions of the bikes. *DataMiner* implements the proposed estimation algorithm (explained in the following sections). When there is some significant data to communicate to the user, this is transmitted from the server side to the app via the *Communicator* component.

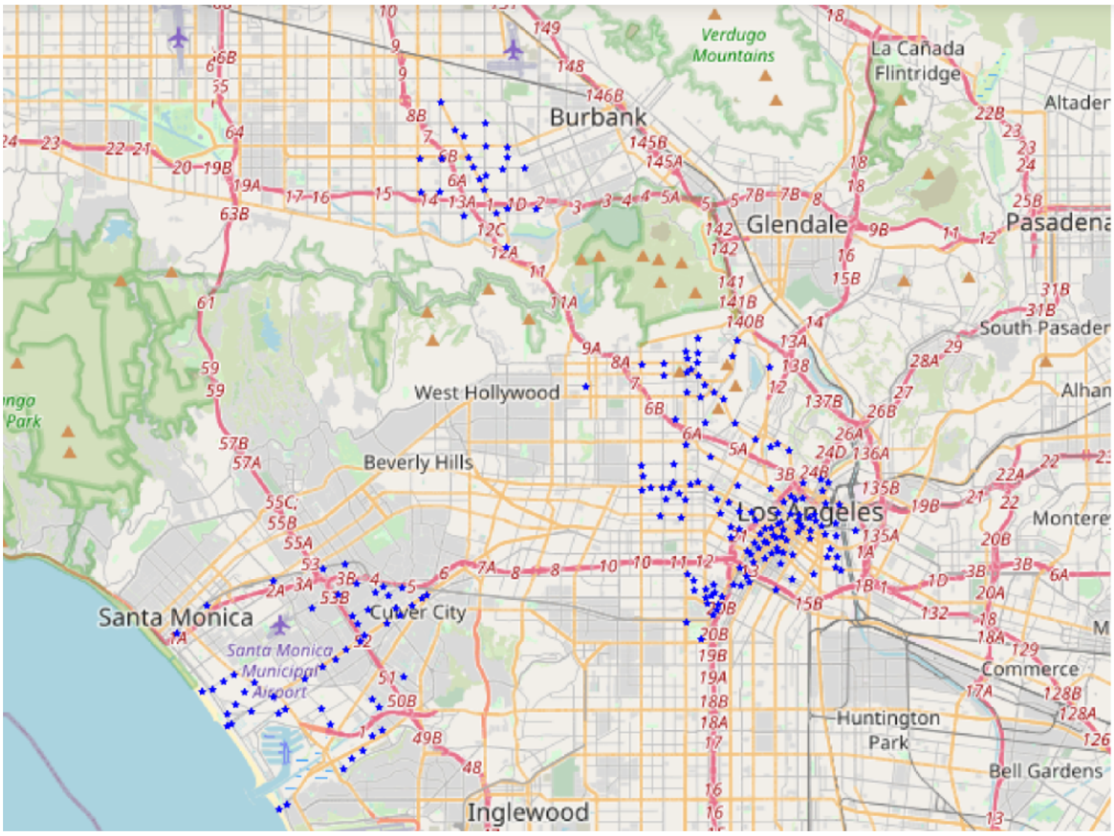

Figure 3: Bike stations are represented by blue stars and relate to trips recorded in the period April-June 2021. [Google Maps, accessed July, 2021]

4. Dataset

The dataset used in our experiments to perform tests was provided by Metro Bike Share[2]. Cyclist trips in Los Angeles, California, have been collected since July 7, 2016, and this project includes recordings for the following 20 quarters. The file of each period, for each line, includes: a numeric travel identifier, travel duration in minutes, travel start date and time, start station name with its geographical position (latitude, longitude), end station name and its geographical position (latitude, longitude), and the numeric identifier of the bike. Moreover, there are pass subscription details, and bike type.

The position of each person who uses the bike sharing service is identified each time she uses the card in the bike parking stations. User privacy is guaranteed since a numeric identifier is associated with each bike used. Journeys lasting less than one minute and trial journeys are not retained. At present the dataset corresponds to $1,252,386$ registered trips of $208,000$ users, for a total of $4,091,563$ miles.

[2]https://bikeshare.metro.net/about/data/

5. Methodology

In this section we present a description of FP-Growth, a data mining algorithm proposed by Han et al. [21, 10]. We highlight that it is our novel proposed idea to apply it for the geo-spatial field in order to find common paths using an algorithm having a short execution time. Then as a further step in our proposal we predict future movements.

The common purpose of this and other data mining algorithms, and among the most interesting we mention Apriori [22], is to extract frequent elements or tuples of elements from a set of transactions and the association rules from large datasets. It is possible to find more details of the application of Apriori algorithm in order to find common paths in [23], and as far as the prediction of movements in a discretized spatial area in [6]. A disadvantage of this strategy, however, is that the count of the *support* of an element (i.e. the number of transactions it is present in) requires scanning the dataset over and over again, and this greatly affects the execution time.

The main difference that led us to opt for the FP-Growth algorithm is that the generation of frequent candidates is not necessary for it. This was done in Apriori at each iteration, that is, every time an element of greater size than that of the previous level was investigated. FP-Growth works on a tree representation of the starting dataset, therefore, thanks to this more compact storage, managing large volumes of data is no longer a problem. Memory errors due to large occurrence tables, necessary for Apriori processing, are thus overcome.

Each set of elements is mapped into a specific path of the tree and the efficiency of this algorithm derives from its "divide et impera" method. The initial part of this approach is to find the support of each element, in our case, to determine which stations are shared by a chosen minimum number of users. After finding these frequent points, they are sorted in descending order according to their support and associated with a branch of the tree. Figures 4 and 5 show how FP-Growth works using a small-sized example.

The tree is then updated: (i) if an element already exists in it, its count is increased, otherwise

Transaction	Itemset
1	I_F, I_A, I_C, I_D, I_G, I_M, I_P
2	I_A, I_B, I_C, I_F, I_L, I_M, I_O
3	I_B, I_F, I_H, I_O
4	I_B, I_K, I_C, I_P
5	I_A, I_F, I_C, I_L, I_P, I_M, I_N

Item	Support
I_A	3
I_B	3
I_C	4
I_D	1
I_F	4
I_G	1
I_H	1
I_K	1
I_L	2
I_M	3
I_N	1
I_O	2
I_P	3

MINIMUM SUPPORT = 3

Item	Support
I_A	3
I_B	3
I_C	4
~~I_D~~	~~1~~
I_F	4
~~I_G~~	~~1~~
~~I_H~~	~~1~~
~~I_K~~	~~1~~
~~I_L~~	~~2~~
I_M	3
~~I_N~~	~~1~~
~~I_O~~	~~2~~
I_P	3

Figure 4: How FP-Growth works using a small example: on the left a list of transactions and their items, in the centre a list of items and its support, on the right the selection of items having a minimum given value of the support.

Frequent item	Support
I_F	4
I_C	4
I_A	3
I_B	3
I_M	3
I_P	3

Transaction	Itemset
1	I_F , I_A, I_C, I_D, I_G, I_M, I_P
2	I_A, I_B, I_C, I_F, I_L , I_M , I_O
3	I_B, I_F, I_H , I_O
4	I_B, I_K, I_C , I_P
5	I_A, I_F, I_C, I_L , I_P, I_M, I_N

Transaction	Ordered itemset, with support ≥ 3
1	I_F, I_C, I_A, I_M, I_P
2	I_F, I_C, I_A, I_B, I_M
3	I_F, I_B
4	I_C , I_B, I_P
5	I_F, I_C, I_A, I_M, I_P

Figure 5: Given the example shown in Figure 4, the right table shows the transactions having the items whose support is at least a given value.

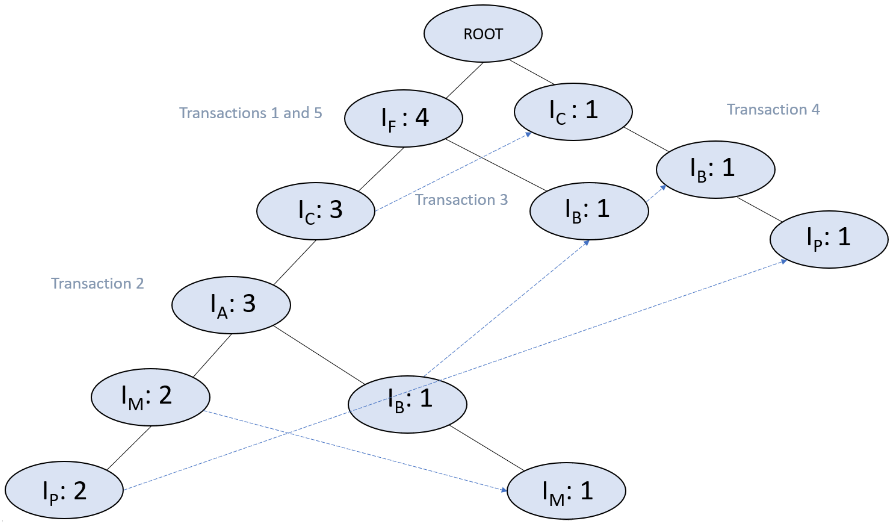

Figure 6: The FP-Tree of the example in Figures 4 and 5.

(ii) a new branch is created and connected to the parent node. To find a common path, starting from an element of the root, we scroll along a branch of the tree. Further examples and details of FP-Growth, even though not applied to the geo-spatial context, can be found in [24].

Figure 6 shows the final FP-Tree of the previous example. In this case the frequent itemsets extracted are: I_F -I_C -I_A-I_M with support $= 3$, and all its subsets with support 3, except $I_F = 4$; I_C -I_P with support $= 3$, I_C with support 4, I_P with support $= 3$ and I_B with support $= 3$.

Note that the FP-Growth runtime increases linearly with the number of elements, while in Apriori this growth is exponential. The frequent patterns in Apriori are obtained after the execution of all the iterations, while with FP-Growth it is possible to obtain them starting from the root-element and scrolling along a single branch. The dataset scan with FP-Growth method is done only twice, as opposed to Apriori in which the scan is done as many times as the number of iterations. Sorting by decreasing frequency allows a faster execution time than for the increasing one.

We can say that FP-Growth is the best strategy for our goal, which allows us to effectively find common routes among the same group of cyclists in a large (spatial) dataset. According to our approach, in fact, it is not necessary to make a comparison of trips (sequences of stops in bike parking) for all pairs of users. Our experiments show that this method is about 6 times faster than Apriori.

6. Results

This section shows the results of our tests. For the bike dataset, in the most recent quarter (April-June 2021) the data consist of $59,081$ routes, $2,824$ cyclists traveling between 215 bike start stations and 215 end stations, and these are shown in the map drawn in Figure 3.

By applying the FP-Growth algorithm in order to find the routes connecting two "hot" stations, we initially set a high support: corresponding to 10% of cyclists in different time slots. The result showed us 8 pairs of common stations shared by a minimum of 282 cyclists to a maximum of 442. The execution time of the FP-Growth algorithm was $40\ ms$ on an Intel Core i5 at 1GHz having 16 GB of RAM: all the tests were performed in Python 3 language.

By lowering the threshold of number of users, for a support equal to 100 bike IDs, 153 different routes were detected in a run time of $116\ ms$. To find the small groups of people moving together, we set the minimum support as 5 by running the algorithm on the columns of the dataset corresponding to: same departure station, same arrival station, same departure time and same end time. The test showed groups of 5 to 7 cyclists who shared 13 routes in the same time slot (9am-12pm). Since the exact position of cyclists in the intermediate roads between two stations is not known, to verify that the common routes are plausible we checked the value in the "duration trips" column, which was compatible for the whole group that was moving together.

Even in this case the algorithm gives the output in a short time, i.e $102\ ms$ for FP-Growth, while running Apriori takes $633\ ms$. This gain of the execution time is very important, especially if you want to apply this method to a longer period of time or to give statistics in near real time.

In addition, when using Apriori to analyse a larger set of paths, some memory errors occurred, both for searching for frequent patterns and for predictions using association rules. When using FP-Growth algorithm, we can generate a sequence of multiple paths, called *association rules*. On the basis of the experiments we deduce that a group will move from point A to point B and then to other positions, with a certain probability. This is called *confidence* and indicates the conditional probability of being in B if you have previously stopped in A.

By setting a minimum threshold of 60% we can determine which stations the cyclists will arrive at in the near future. To give an example, the tests show that starting from the time slot $9-12$ a minimum group of 10 users will move from the "Normandie & Hollywood" bike station to "Figueroa & Cesar Chavez" with Confidence 78%, and the reverse route with Confidence 70%; the group will follow the path " Pacific & North Venice" $\rightarrow$ "Ocean Front Walk & North Venice" with 61% Confidence; and finally "Ocean Front Walk & North Venice" - "Ocean Front Walk & Navy" with probability 60% (round trip). Figure 7 shows the union of the predicted routes, assuming that the group follows the shortest and most comfortable route to reach all the stages.

For the above prediction experiments, 70% of the available data was used for training. The

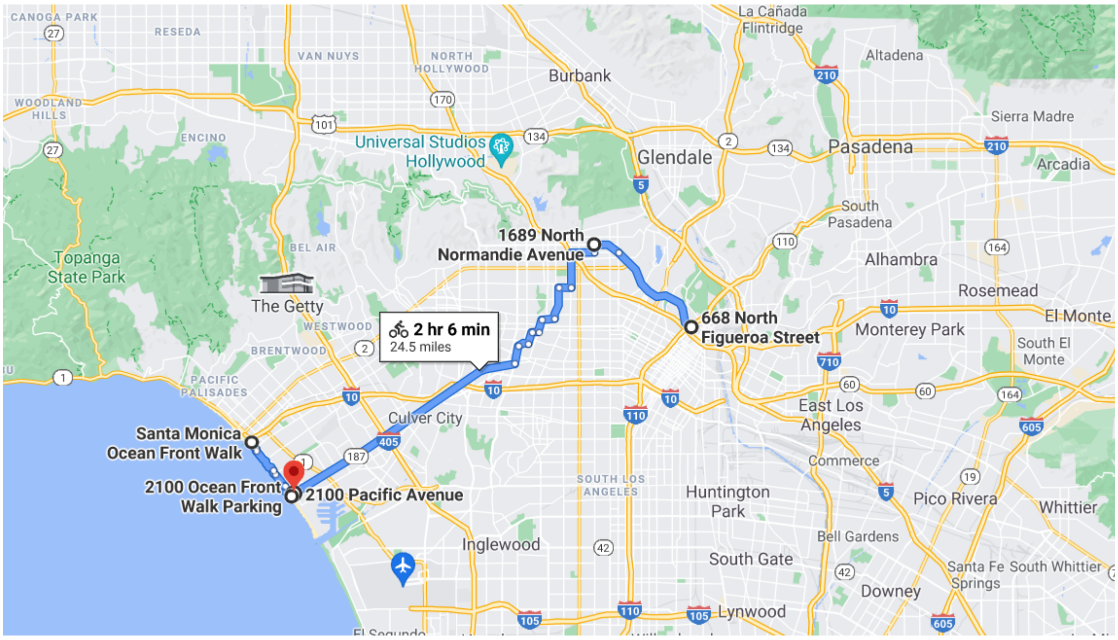

Figure 7: Route prediction for the same group of cyclists. [Google Maps, accessed July, 2021]

remaining 30% of the data was referred to as the validation and test, useful for verifying the results obtained.

7. Concluding Remarks

This paper has shown the feasibility of the proposed approach providing means to enhance a bike sharing service by alerting interested people of bike availability at their preferred bike stations.

Firstly, users are provided with assistant agents as an app on their smartphone that gathers the geographical coordinates when using a shared bike service. Moreover, the app lets users express their preferences when wishing to rent a bike. Secondly, the FP-Growth algorithm has been put at work, and executes on a server to analyse gathered data to provide the said estimates. Such an algorithm has not been used on locations data before.

The experiments have shown that the proposed approach is very fast and then suitable to process the abundance of data available, as it is a linear time algorithm. We have processed a large dataset comprising several thousands of items to recognise common paths of bikes and make estimates of future stops for bikers in a few milliseconds.

Users wishing to use bike sharing services could then use the proposed app to have an estimate of bikes available in a future time slot in a bike station. Then, they could reserve a bike, and be alerted when the bike arrives, or whether the estimated time for availability is updated. Being an estimate availability for the future, though with high probability, sometimes the availability would be cancelled, however the user would be notified beforehand. We believe that the proposed approach, by letting users plan beforehand their transport means, and schedule, would provide users with a happier experience.

In order to take into account the effects on the use of bikes of the seasonal variability of the weather, as well as changes in people preferences, due to the beginning of the academic year, etc., we consider the most recent data for our statistics, when enough data are gathered to gain sufficient confidence on estimates.

The proposed approach and the implemented solution could be applied to other data related to geographical positions when needing estimates of future locations, rather than bikes. E.g. the movements of electric kick scooters, being very numerous, future drones, and when considering data characterised by GPS positions having fine granularity, even when different kinds of vehicles are considered, could make use of the proposed approach as the linear complexity of the algorithm makes it possible to be applied for a growing amount of data.

8. Acknowledgments

The authors acknowledge project TEAMS funded by University of Catania PIACERI 2020/22.

References

[1] 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.

[2] M. Aliannejadi, F. Crestani, Personalized context-aware point of interest recommendation, ACM Transactions on Information Systems (TOIS) 36 (2018) 45.

[3] Y.-L. Hsueh, H.-M. Huang, Personalized itinerary recommendation with time constraints using gps datasets, Knowledge and Information Systems 60 (2019) 523–544.

[4] 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.

[5] F. de Nijs, G. Theocharous, N. Vlassis, M. M. de Weerdt, M. T. J. Spaan, Capacity-aware sequential recommendations, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18, 2018, p. 416–424.

[6] C. Cavallaro, G. Verga, E. Tramontana, O. Muscato, Suggesting just enough (un)crowded routes and destinations, in: Proceedings of 21st Workshop "From Objects to Agents" (WOA), Bologna, Italy, 14–16 September2020, volume 2706, 2020, pp. 237–251. URL: http://ceur-ws.org/Vol-2706/.

[7] D. Quercia, R. Schifanella, L. M. Aiello, The shortest path to happiness, in: Proceedings of the 25th ACM conference on Hypertext and social media, ACM, 2014. doi:10.1145/2631775.2631799.

[8] 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.
[9] M. Mythili, A. M. Shanavas, Performance evaluation of apriori and fp-growth algorithms, International Journal of Computer Applications 79 (2013).
[10] C. Borgelt, An implementation of the fp-growth algorithm, in: Proceedings of 1st international workshop on open source data mining: frequent pattern mining implementations, 2005, pp. 1–5.
[11] R. Pensa, A. Monreale, F. Pinelli, D. Pedreschi, Pattern-preserving k-anonymization of sequences and its application to mobility data mining, in: Proceedings of Privaci in Location-Based Applications (PiLBA), volume 397, 2008.
[12] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth, in: Proceedings of 17th International Conference on Data Engineering, IEEE Comput. Soc, 2004. doi:10.1109/icde.2001.914830.
[13] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, Dbscan revisited, revisited: why and how you should (still) use dbscan, ACM Transactions on Database Systems (TODS) 42 (2017) 19.
[14] 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.
[15] B. Ludwig, B. Zenker, J. Schrader, Recommendation of personalized routes with public transport connections, in: Proceedings of Intelligent Interactive Assistance and Mobile Multimedia Computing, Springer, 2009, pp. 97–107.
[16] R. Shourouni, M. Malek, and, Route recommendation based on local users' trajectories, Journal of Geospatial Information Technology 4 (2017) 53–67. doi:10.29252/jgit.4.4.53.
[17] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM 46 (1999) 604–632. doi:10.1145/324133.324140.
[18] A. Cecaj, M. Mamei, Investigating economic activity concentration patterns of co-agglomerations through association rule mining, Journal of Ambient Intelligence and Humanized Computing 10 (2017) 463–476. doi:10.1007/s12652-017-0665-3.
[19] J. Heaton, Comparing dataset characteristics that favor the Apriori, Eclat or FP-growth frequent itemset mining algorithms, in: Proceedings of SoutheastCon, IEEE, 2016. doi:10.1109/secon.2016.7506659.
[20] E. Tramontana, G. Verga, Mitigating privacy-related risks for android users, in: Proceedings of IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2019.
[21] J. Han, J. Pei, Y. Yin, R. Mao, Mining frequent patterns without candidate generation: A frequent-pattern tree approach, Data Mining and Knowledge Discovery 8 (2004) 53–87. doi:10.1023/b:dami.0000005258.31418.83.
[22] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of 20th International Conference on Very Large Data Bases, VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, p. 487–499.
[23] C. Cavallaro, J. Vitrià, Corridor detection from large GPS trajectories datasets, Applied

Sciences 10 (2020) 5003. doi:10.3390/app10145003.

[24] P. Fournier-Viger, J. C.-W. Lin, B. Vo, T. T. Chi, J. Zhang, H. B. Le, A survey of itemset mining, WIREs Data Mining and Knowledge Discovery 7 (2017) e1207. doi:https://doi.org/10.1002/widm.1207.