User Assistance for Predicting the Availability of Bikes at Bike Stations

Claudia Cavallaro[a], Emiliano Tramontana[b]

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

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

Abstract

Assistant agents have been developed 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. Assistant agents giving such an alert are provided to users as an app on their smartphone.

Keywords

Assistant agents, Recommendation system, Bike-sharing

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. E.g. it is possible to determine the most visited tourist places in a city and the usual paths traversed to reach such places [1].

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 timeframe [2], paths inducing some kinds of emotions [3, 4], 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. Such an app can be seen as an *agent* assisting the user in her daily routine.

In previous studies, the analysis of GPS coordinates has been performed using several techniques, such as computing the point of interests, and then determining how frequent each user can be found nearby. The Apriori algorithm has been used to find how frequently a set of

22nd Workshop "From Objects to Agents", September 01–03, 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)

places have been visited by users passing through the same path [2]. However, the execution of the Apriori algorithm takes a long time [5]. 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 [5], 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

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 [6], Pensa et al. address 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 measure the similarity of each pair of patterns as support density and length. By using the technique PrefixSpan [7], ROI's sequences have been represented as a tree, where each node is a ROI. Then, each of branch of the tree was associated with the support in each trajectory and subtrees having infrequent ROIs were removed. Finally, they present the framework P2kA that anonymizes the dataset and thus preserves the privacy of users in the frequent paths extracted. Unlike our proposal, in [6] the objective is not to make predictions on future destinations, while our strategy 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, were used for the purpose of identifying parts of common routes or shared Points Of Interest [1]. In [8], Crociani et al. offer an unsupervised

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

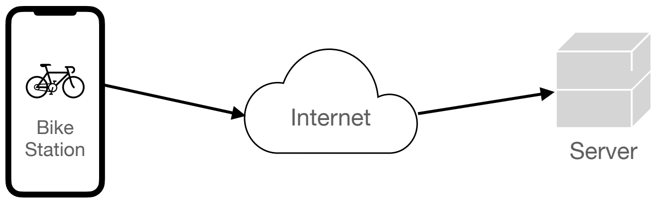

Figure 1: Software architecture for the multi-agent system.

learning approach for an automatic lane detection in multidirectional pedestrian flows, taking into account the angular distance during the movement.

Regarding recommendation systems, in [3], of Quercia et al. gather metadata from the pictures in Flickr, and determine routes to be suggested according to parameters of beauty, quietness and happiness. They compute the probability that an individual visits a certain destination because it is pleasant. In the literature other algorithms, such as the one presented in [9], recommend 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 is presented in [10]: the authors use HITS [11] algorithm, and it is based on the travel experiences of the users (hub scores) and the interests of the road segments (authority scores).

In [12], Cecaj and Mamei present a useful way to extract associative rules, with the Eclat algorithm, to find co-locations of companies in industrial agglomerations. Our approach mine 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 [13].

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 his preferences and requests. Figure 1 gives an overview of the software architecture.

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 2 shows some of the 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

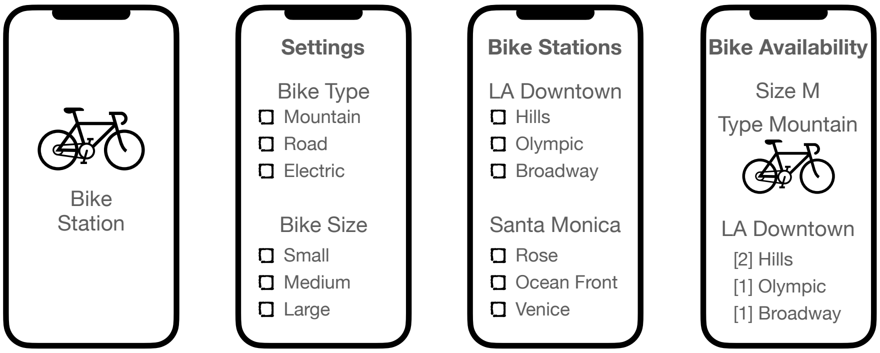

Figure 2: User interface provided by the app on the smartphone by the assistant agent.

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.

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.

4. Dataset

The dataset used in our experiments to perform tests was provided by Metro Bike Share, and it is available on https://bikeshare.metro.net/about/data/. Cyclist trips in Los Angeles, California, have been collected since July 7, 2016, and this project includes recordings for the next 20 quarters. The file of each period includes, for each line, 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 also the numeric identifier of the bike. There is also other information not relevant to our analysis, such as pass subscription details or bike type. The position of each person who uses the bike sharing service is identified each time he/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.

5. Methodology

In this section we present a description of FP-Growth, a known data mining algorithm proposed by Han et al. in [14]. We highlight that it is our novel proposed idea to apply it for the geo-spatial field in order to find common paths in short execution time, and to predict future movements.

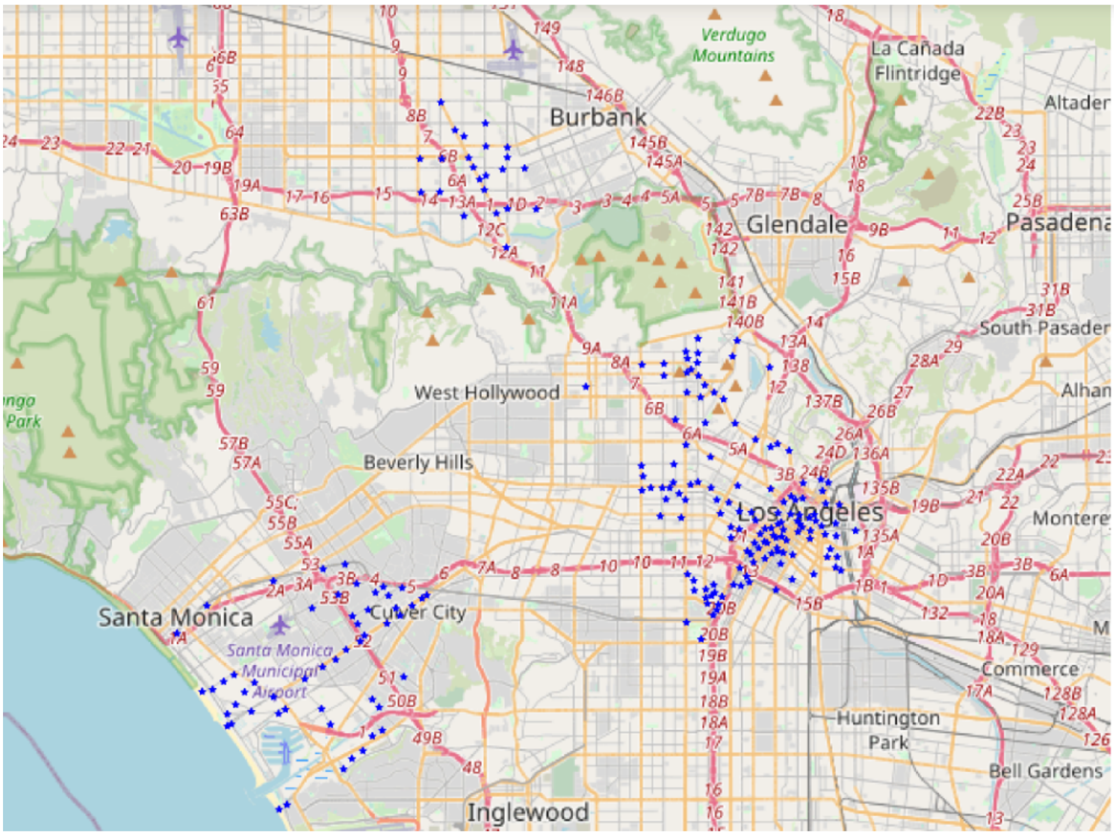

Figure 3: Bike stations, represented by blue stars, relating to trips recorded in the period April-June 2021.

The common purpose of this and other data mining algorithms, and among the most interesting we mention Apriori [15], 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 [16], and as far as the prediction of movements in a discretized spatial area in [2]. 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.

The tree is then updated: if an element already exists in it, its count is increased, otherwise 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 if not applied to the geo-spatial context, can be found in [17].

We can also add 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 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

In this section the results of our tests are shown; referring for example to the last quarter (April-June 2021) the data consist of $59,081$ routes, $2,824$ cyclists traveling between 215 bike start stations and 215 end stations, shown in the map 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 in detail 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 (9-12 A.M.). 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.

Also in this case the algorithm shows the output in short times: 102 *ms* for FP-Growth, while Apriori 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 Apriori analyzes a larger set of paths returned memory errors, both for

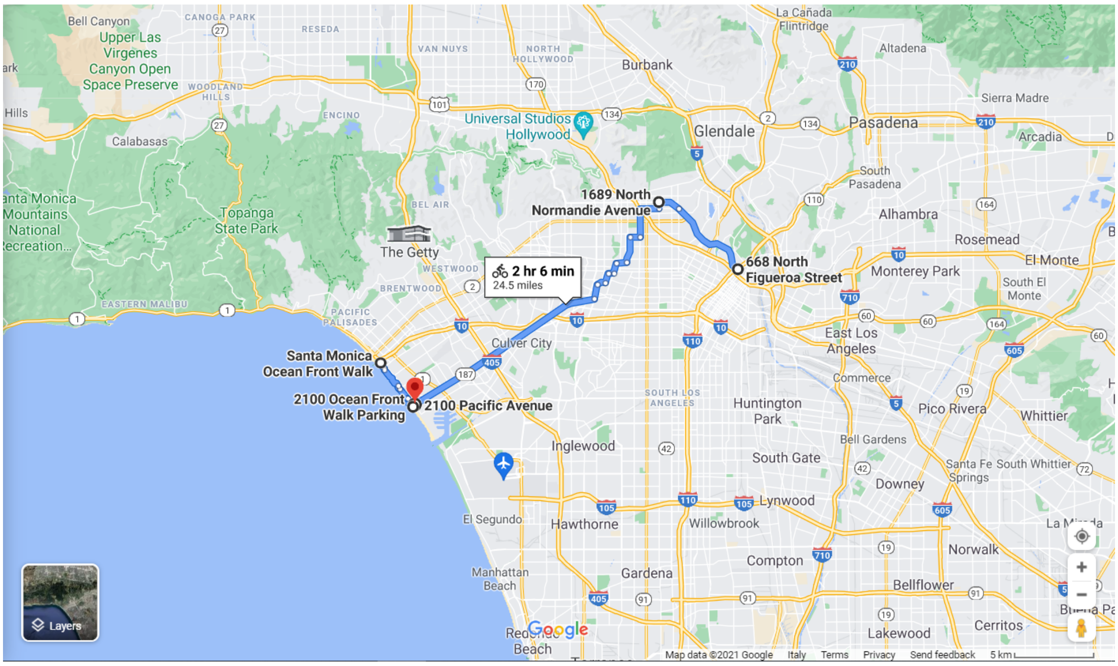

Figure 4: Route prediction for the same group of cyclists.

searching for frequent patterns and for predicting using association rules. Moving on to the latter, let's show what we have achieved with the FP-Growth. The algorithm generates 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 so on 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 know 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 "Figueroa & Cesar Chavez" bike station to "Normandie & Hollywood" with Confidence 78%, and the reverse route with Confidence 70%; it will follow the path "Ocean Front Walk & North Venice" - "Ocean Front Walk & Navy" with probability 60% (round trip) and finally "Ocean Front Walk & North Venice" ->" Pacific & North Venice" with 61% Confidence. Figure 4 shows the union of the predicted routes, assuming that the group follows the shortest and most comfortable route to reach all the stages.

7. Conclusions

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, assistant agents are provided as an app on a smartphone gathering the geographical coordinates of people using a shared bike service, and letting users express their preferences when wishing to rent a bike. Secondly, the FP-Growth algorithm has been put at work, and executing 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.

Users wishing to use bike sharing services can then plan beforehand their transport means, schedule, and have a happier experience.

References

[1] 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.

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

[3] 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.

[4] C. Berzi, A. Gorrini, G. Vizzari, Mining the social media data for a bottom-up evaluation of walkability, in: International Conference on Traffic and Granular Flow, Springer, 2017, pp. 167–175.

[5] M. Mythili, A. M. Shanavas, Performance evaluation of apriori and fp-growth algorithms, International Journal of Computer Applications 79 (2013).

[6] R. Pensa, A. Monreale, F. Pinelli, D. Pedreschi, Pattern-preserving k-anonymization of sequences and its application to mobil- ity data mining, in: PiLBA, volume 397, 2008.

[7] 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 17th International Conference on Data Engineering, IEEE Comput. Soc, 2004. doi:10.1109/icde.2001.914830.

[8] 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.

[9] B. Ludwig, B. Zenker, J. Schrader, Recommendation of personalized routes with public transport connections, in: D. Tavangarian, T. Kirste, D. Timmermann, U. Lucke, D. Versick (Eds.), Intelligent Interactive Assistance and Mobile Multimedia Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 97–107.

[10] 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.

[11] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM 46 (1999) 604–632. doi:10.1145/324133.324140.

[12] 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.

[13] J. Heaton, Comparing dataset characteristics that favor the Apriori, Eclat or FP-growth frequent itemset mining algorithms, in: SoutheastCon 2016, IEEE, 2016. doi:10.1109/secon.2016.7506659.

[14] 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.

[15] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, p. 487–499.

[16] C. Cavallaro, J. Vitrià, Corridor detection from large GPS trajectories datasets, Applied Sciences 10 (2020) 5003. doi:10.3390/app10145003.

[17] 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.