Clustering User Trajectories to Find Patterns for Social Interaction Applications *

Reinaldo Bezerra Braga[1], Ali Tahir[2], Michela Bertolotto[2], and Hervé Martin[1]

[1] LIG UMR 5217, UJF-Grenoble 1, Grenoble-INP, UPMF-Grenoble 2, CNRS
38400, Grenoble, France
(braga, herve.martin)@imag.fr

[2] School of Computer Science and Informatics, University College Dublin (UCD)
Dublin, Ireland
(ali.tahir, michela.bertolotto)@ucd.ie

Abstract. Sharing of user data has substantially increased over the past few years facilitated by sophisticated Web and mobile applications, including social networks. For instance, users can easily register their trajectories over time based on their daily trips captured with GPS receivers as well as share and relate them with trajectories of other users. Analyzing user trajectories over time can reveal habits and preferences. This information can be used to recommend content to single users or to group users together based on similar trajectories and/or preferences. Recording GPS tracks generates very large amounts of data. Therefore clustering algorithms are required to efficiently analyze such data. In this paper, we focus on investigating ways of efficiently analyzing user trajectories and extracting user preferences from them. We demonstrate an algorithm for clustering user GPS trajectories. In addition, we propose an algorithm to correlate trajectories based on near points between two or more users. The obtained results provided interesting avenues for exploring Location-based Social Network (LBSN) applications.

1 Introduction

Social network platforms have emerged as a collaborative solution to provide social connectivity, giving people the capability to create virtual communities and share interests, opinions, and personal information with other users. However, while there has been an increase in virtual communities, a reduction of social interactions in real communities is evident. We have noticed that social network platforms do not make use of correct context-aware mechanisms in order to improve social contacts in real communities. Therefore, we argue that these

* Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland (SFI) under the National Development Plan, the IRCSET Ulysses program, French Ministry of Higher Education and Research, ÉGIDE program and European Cooperation in Science and Technology (COST). The authors gratefully acknowledge this support.

platforms should be based on users' daily routines to increase social interactions among mobile users in real communities.

Nowadays, we have observed a large adoption of smart phones and social networks. As a consequence several mobile social applications have been developed to register social behaviors of mobile users [1] including Ipoki[3], Google Latitude[4], Carticipate[5] and Daily Places[6]. Despite the availability of these mobile social applications to register and share users' daily routines, we face a rapid increase of diverse kinds of space-associated data, such as measurements from mobile sensors, GPS tracks, or georeferenced multimedia. As prospective sources of useful knowledge and information, these data require scalable methods of analysis, which need to consider the particular attributes of the geographical space, such as heterogeneity, diversity of characteristics and relationships, spatio-temporal autocorrelation, and multiple map scales.

Furthermore, recording GPS tracks generates a large amount of data. This data holds spatio-temporal information about a moving object (such as pedestrians, cars, buses, etc.). In order to analyze such data there exists several exploratory as well as data mining techniques. Clustering and aggregation (data mining) techniques have generally been adopted to explore and analyze movement data when visualization (exploratory) techniques are not enough to explore large spatio-temporal datasets. This scenario is also pertinent in case of LBSN applications we have developed.

The purpose of this paper is to explore the capabilities provided by clustering algorithms to analyze user trajectories and extract relevant information from them. We have focused on clustering and aggregating multiples trajectories generated by the same user in order to identify his/her preferences. Once each user preference is identified we apply trajectory correlation algorithm in order to find similarities between multiple user trajectories and near points of interests between two or more users.

To validate our approach, we considered a dataset of trajectories representing a user daily routine (i.e. to go from home to work). We implemented and tested the clustering and trajectory correlation algorithm to understand similarities between users. The results show that our technique is effective in analyzing trajectories datasets and extracting the user preferences. Besides that, the correlation trajectory algorithm is able to effectively find similar PoI between two or more users. Based on the results we envision interesting avenues for social interactions between users.

The rest of this article is organized as follows. To provide the necessary context for our work, we start with the related work in the next section. The proposed architecture, clustering and correlation algorithms are described in Section 3. Section 4 shows experimental results and evaluation we have conducted. Finally, Section 5 presents the conclusions and some directions for future work.

---

[3] ipoki.com

[4] google.com/latitude

[5] carticipate.com

[6] dailyplaces.com

2 Related Work

In general, mobile social applications that implement Mobile Trajectory Based Service (MTBS) consider information about time and space to represent users' trajectories in transportation networks. In [2], the authors present a new strategy to find the fastest route in dynamic transportation networks, making use of previous trajectory information and real-time traffic conditions. Other strategies use the Dijkstra algorithm to solve the same problem in dynamic networks [3]. An important work was proposed in [4], in which the authors introduce a mechanism to model the intelligence of taxi drivers and the properties of dynamic networks to find the fastest route. All these strategies allow the sharing of mobile traces or trajectories to provide a large number of mobile social applications, ranging from a simple navigation mechanism to a robust context-aware and trust-based recommendation system [5].

In spite of the large number of mobile social applications based on context aware information and the adoption of several social networks, some studies show that virtual communities do not increase significantly the amount of social interactions in real communities [6] [7]. Social interactions in the form of user trajectories can generate a huge amount of spatio-temporal data. This can be roughly categorized into a single as well as multiple users trajectories. The former relates to users generating their trajectories over a certain time period, while the latter focuses on group of users interacting socially with their friends and generating their trajectories. In both cases the amount of trajectories produced could be enormous and therefore challenging to interpret for the analysts. Many techniques exist in the literature, however clustering and aggregation techniques are found to be the most suitable for such analysis.

Clustering is a data-mining technique to identify similar and dissimilar groups in a given dataset. The clustering methods however can be classified broadly into partitioning, hierarchical, density-based, grid-based, model-based, constrain-based methods and clustering high-dimensional data [8]. While the overall objective of clustering is the same, they differ based on how they analyze additional parameters such as outliers, noise analysis and dimensions of a given dataset. Each technique can be described in detail with their merits and de-merits. One such study evaluated clustering techniques with focus on trajectory clustering [9].

In our scenario of social interaction application the focus is to find groups with varying density and concentration. For this purpose, density-based clusters are found to be suitable. The main idea is to enlarge a cluster as long as the density of data objects in the neighborhood exceeds a certain threshold value. A typical condition is that for each data point within a cluster, the neighborhood of a given radius has to contain at least a minimum number of points. These methods are quite efficient to find noise and outliers as well as to discover clusters of arbitrary shape. When trajectories are collected in real time, they usually suffer low resolutions of measurements, which make noise tolerance a highly considerable feature [10]. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [11] and Ordering Points To Identify the Clustering Structure (OPTICS) [12] are widely used density-based clustering methods.

OPTICS has proven effective when offered to trajectory data in some applications [13]. The approach successfully clustered mouse trajectories and obtained good results. An important input to clustering algorithm is an appropriate distance metric. Morris and Trivedi [9] performed an evaluation and discussed distance similarity measures based on fixed length measures (such as Hu Euclidean and PCA subspace) as well as time-normalized distances (also suitable for unequal trajectories length) such as Dynamic Time Warping (DTW), Longest Common Subsequence (LCS) and Modified Hausdorff (MH) (see [9] for an overview of these measures). In case of trajectory data few distance measures have been provided by CommonGIS, a stand-alone visualization tool [14] developed for analysis of movement datasets. They have defined two simple distance methods namely common start and common destination to group trajectories based on their starting and ending points respectively. They also defined two more complex functions called route similarity and route similarity and dynamics. These methods compare two trajectories of unequal length and find the spatial as well as spatio-temporal distance between two trajectories.

3 Our Approach

We present a novel solution in order to increase social interactions by relating daily routines and points of interest based on trajectories of mobile users. For instance, a mobile social application jointly with a social network can answer the following questions: Which of my friends stop in my preferred bakery at the same period of the day? Do any of my friends pass near my apartment to get from their home to their work? Which of my contacts will be passing close to me during the week?[7]

In relation to this we introduce the following 3 definitions to support our discussion.

1. **Road Segment (S)** is defined as a directed link between two extreme points (s_a) and (s_b), composed by a list of intermediate points by using a polyline.
2. **User Trajectory (U_T)** is defined as a set of road segments. Thus, $U_T = \{S_1, S_2, S_3...S_n\}$, where the end point of S_k is the point just before the start point of S_{k+1}, and $(1 \leq k < n)$.
3. **Trajectory (T)** is defined as a set of consecutive points captured through a Global Positioning System (GPS) to one travel performed by the user. Each position (p) is composed of a set of information (latitude, longitude, altitude, direction, time stamp for each registered point (t_p) and an approximate speed provided by the GPS). Since $T = \{p_1, p_2, p_3, ..., p_n\}$, the time interval between two points is computed by the subtraction of $t_{p(k+1)} - t_{p(k)}$, where $(1 \leq k < n)$. Although the points are characterized by latitude, longitude and altitude, we focus on points in 2D space (latitude and longitude) to represent the position of each user.

[7] The user defines the contacts to share his/her daily routine.

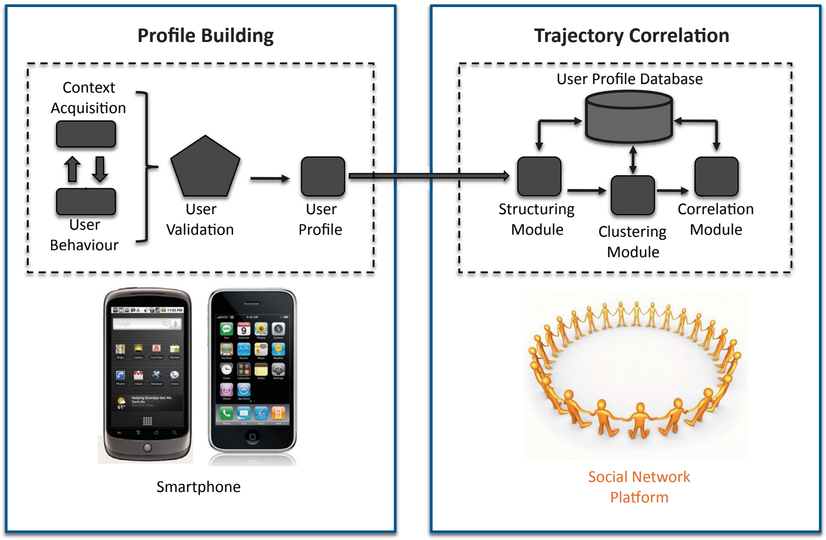

Fig. 1. Architecture overview.

Two major components compose our architecture: Profile Building and Trajectory Correlation. Figure 1 presents this architecture. The profile building component should operate in offline mode, but the trajectory correlation components works in online mode. The offline part only needs to be performed once unless the trajectory archive is updated.

As we can observe in the profile building process, users can use a mobile social application to register their trajectory in order to describe their daily routine. After visualizing and validating the trajectory that represents users' daily routine, the user profile is created and the trajectory information is sent to the next step, the structuring module. At this moment, the structuring module verifies if there is a previous trajectory for the same user stored in the database. If there is no trajectory, it creates a new user's daily routine. On the other hand, if multiple trajectories are found, clustering and aggregation techniques can support the analysis to identify the aggregated trajectory (a best representative of user's daily routine). The user daily routine then is enriched with additional information about Points of Interest. Finally, the structuring module exports the enriched information to update the user profile database. These two components are detailed in the next sections.

3.1 Profile Building

The user profile can be designed taking into account two basic types of data that are used for constructing and enriching the profile model. These two basic types are defined as *personal* and *contextual* data. Personal data describes the main details of an entity and the contextual data characterizes the situation. An entity can be a person, place, physical or computational object. For example, in a personal tracking application for mobile users, the personal data would be the information about the user, such as name, birthday, gender, etc. On the other hand, contextual data would be composed of movement records that the user performed over a period of time. A movement record can include such characteristics as the initial point, speed, direction, and time, as well as weather information. We define an entity as a mobile user using a smart phone equipped with GPS, digital camera and Internet connection (e.g. 3G or Edge).

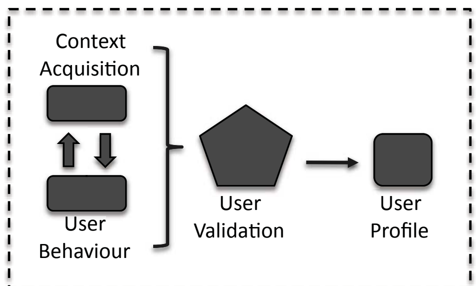

Fig. 2. The profile building process.

In addition, we use a third type of data, which is named *behavioral* data. Behavioral data is defined according to specifications for representing Points of Interest (POI) of the users, which has been developed by the W3C Points of Interest Working Group Charter [15]. This Working Group has defined specifications for Points of Interest data that can be used in a large number of applications, such as augmented reality browsers, geo-caching and games, mapping and navigation systems, and many others. The behavioral data describes the behavior of the users learned from their daily routines. One way to define user behavior is with a set of conjunctive rules, such as classification or association rules. Some examples of rules describing user behavior are: "When user Hervé goes from work to his residence, he usually stops at the bakery", "Every Monday Carina goes from her work to the tennis court at 13:00 and comes back to her work at 14:30", "Whenever user Reinaldo goes from his residence to his office, he stops in the Residence Matisse at 08:00 to take his friends to work". The use of rules in profiles offers a perceptive, descriptive and modular way to characterize user behavior and was presented in[16].

The rules can be either determined by specialists or derived from transactional data of a user, making use of clustering algorithms or machine learning techniques. Since we consider mobile social applications in the profile building process, our rule discovery method is used individually to the transactional data of each user, capturing and comparing personal behaviors. Hence, the rules are discovered using a clustering algorithm in multiple user trajectories.

3.2 Clustering Algorithm

We have adapted OPTICS [12] clustering algorithm, which produces an ordering of a dataset while storing the core distance and a suitable reachability distance of each user trajectory. OPTICS provides information about the overall clustering structure unlike other method that computes a flat partitioning of data (such as K-means [17]). A brief overview of OPTICS is presented with the help of underlying terminologies. Assume ρ = object from a dataset D, ε = distance threshold, $N\varepsilon\ (\rho)$ = ε-neighborhood of object ρ, *minPts* = natural number, *minPts-distance*(ρ) = distance from ρ to its *minPts* neighbor. The core distance (CD) is defined as:

$$CD = \begin{cases} Undefined, & \text{if } Card(N\varepsilon(\rho)) < minPts \\ minPts\text{-}distance(\rho), & \text{otherwise} \end{cases}$$

Thus, the *core distance* is the smallest distance ε between ρ and an object in its ε-neighborhood such that ρ would be a core object. The *core distance* is *Undefined*, otherwise. For reachability distance, assume ρ and o = objects from a dataset D, $N\varepsilon\ (o)$ = ε-neighborhood of object o, *minPts* = natural number. The reachability distance (RD) of ρ with respect to o is defined as:

$$RD = \begin{cases} \textit{Undefined}, & \text{if } |(N\varepsilon(o))| < minPts \\ \textit{max(core-distance}(o), \textit{distance}(o,\rho)), & \text{otherwise} \end{cases}$$

Thus, the reachability distance of ρ is the smallest distance such that ρ is directly density-reachable from a core object o. Otherwise, if o is not a core object, even at the generating distance ε, the reachability distance of ρ with respect to o is *Undefined*.

OPTICS produces a reachability plot that shows the cluster ordering and the reachability values. The reachability plot gives a graphical view of the structure of the data by providing data independent visualization. From the output plot, clustering can be obtained by choosing an appropriate threshold value of reachability distances. There are automatic techniques available to identify clusters from this plot, which is applicable when the dataset is very large. Figure 3 illustrates cluster ordering with the help of a reachability plot showing valleys to identify potential clusters. Two additional parameters are of significant importance in OPTICS algorithm (maximum distance threshold and minimum number of neighbors). As Ankerst et al.[12] suggest the distance threshold influences the number of clustering levels, which can be seen in a reachability plot. The smaller the distance, the more objects may have undefined reachability distances. Therefore, the clusters with lower density might be less visible and hence this situation should be prevented. Similarly, the larger minimum neighbor value will yield better results.

3.3 Trajectory Correlation

Taking into account the idea to analyze user's daily routines in order to increase the number of social interactions between users, we propose an optimized algorithm based on Minimum Bounding Rectangles (MBR) [18] and the Hausdorff

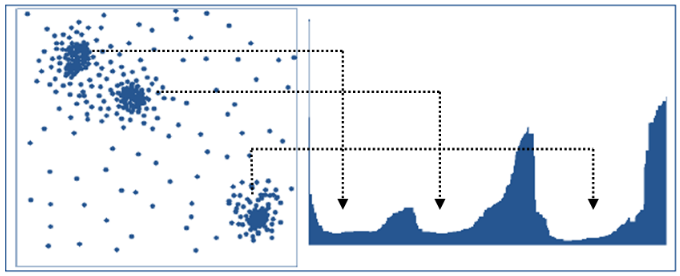

Fig. 3. A reachability plot showing data densities and respective clusters [12]

distance [19]. Firstly, we identify four extreme points of each trajectory (the northernmost, the southernmost, the westernmost and the easternmost). With these points, we create the MBR for the users' trajectories.

The Hausdorff distance is often used to determine the similarity of two shapes [20] and to measure errors for approximating a surface in generating a triangular mesh [21]. In our approach, we are interested to use Hausdorff distance computation in two different cases. Basically, the first case is applied when the algorithm finds a correlated area between two MBRs. It uses Hausdorff distance to compute the distance between the points that are in the correlated area. On the other hand, if there is no correlated area, the Hausdorff distance computation is used to compute the distance of near points between two MBRs. When the distance of two MBRs is found, the algorithm allows the expansion of both MBRs in order to find one or more points of social interactions, taking into account a threshold (D_{max}) for the expansion. The trajectory correlation is executed according to the algorithm as follows.

Algorithm 1 Main algorithm.

if $(Lat_{max(A)} < Lat_{min(B)})$ **or** $(Lat_{max(B)} < Lat_{min(A)})$ **or** $(Lon_{max(A)} < Lon_{min(B)})$ **or** $(Lon_{max(B)} < Lon_{min(A)})$ **then**
  Execute **HausDist** of MBR(A) and MBR(B);
  if HausDist $< D_{max}$ **then**
    Expand MBRs;
  else
    There is no correlated area;
    Stop main algorithm;
  end if
end if
Select correlated area;
Execute **HausDist**;

The Hausdorff distance from MBR(A) to MBR(B) can be determined by exploiting the characteristic that for each MBR face, there has to be at least one object that touches it. Therefore, we identify the face in MBR(A) closest to a face in MBR(B). After that, the algorithm computes the Hausdorff distance of these two faces and compares the result with D_{max}. If Hausdorff distance is less than D_{max}, then both MBRs expand their related faces from the current distance to the result of D_{max}. Once the correlated area of MBRs is found, the main algorithm executes the Hausdorff distance computation of the points.

Our approach is able to identify a correlated area of near points. In addition, it optimizes the Hausdorff distance computation owing to selection of points in the correlated area. This avoids the execution of the distance computation for all points in the trajectory.

Making use of context information, our approach allows the identification of segments S, which can be represented by landmark graphs. This information could be used to increase social interaction. For example, we can capture context information in order to send a message to users, alerting that a friend passes in front of a specific number of the street X all the weekdays between 10:00 AM and 10:30 AM. This message can also contain accurate information of distance, which is acquired by the Hausdorff distance algorithm.

(a) User 1 ($\epsilon = 1000$ & minNbs = 3). (b) User 2 ($\epsilon = 1000$ & minNbs = 3).

Fig. 4. Reachability plots showing clustering structure.

As an additional feature, the trajectory correlation module enables the generation of a message based on the context information. It reads all the fields related to a correlated point in order to automatically create the message that will be sent to one or both users.

4 Results and Discussion

To demonstrate our concept we have applied our approach to two separate users based on their registered trajectories. The overall approach can be summarized in three steps. First of all clustering is applied to individual user trajectories over a period of one month. A typical user route is a trajectory from home to work. After obtaining distinct groups an aggregated trajectory has to be chosen. With the help of visualization and aggregation techniques, a best representa-

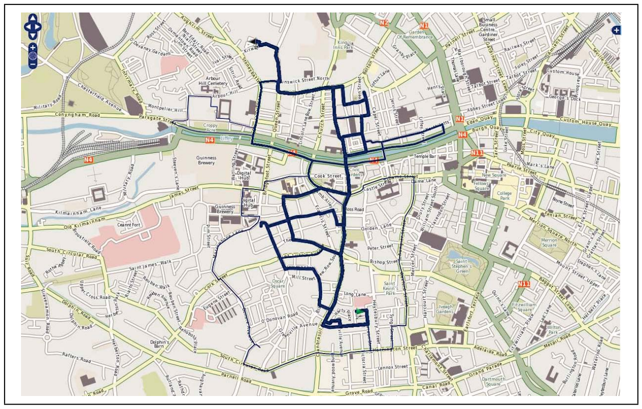

Fig. 5. Three clusters showing distinct routes of User 1 (overlay on map).

tive trajectory for each user is obtained. This aggregated trajectory obtained from several user trajectories is then compared to other users by applying our trajectory correlation algorithm. This will enable groups of users to share similar routes to increase geospatial social interaction. We now explain the different input parameters we have used in order to verify the results.

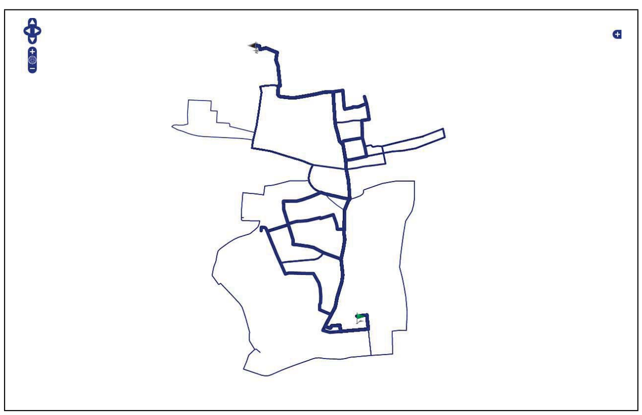

Fig. 6. Three clusters showing distinct routes of User 1 (without overlay).

OPTICS clustering algorithm requires two input parameters: distance threshold (ϵ) and minimum neighbors *(minNbs)*. The authors of OPTICS [12] suggest that the value of these two parameters have to be large enough to yield good results. We structured our experiment in a way that we choose a range of distance threshold values as well as minimum neighbors. For our scenario, we defined the distance threshold between 1000 meters and 15000 meters $\Rightarrow$ *(1000 $\leq \epsilon \leq$ 15000)*. Similarly, for minimum neighbors we selected a value of 1 up to 10 $\Rightarrow$

(1 ≤ minNbs ≤ 10). The experiment was run with a combination of values for

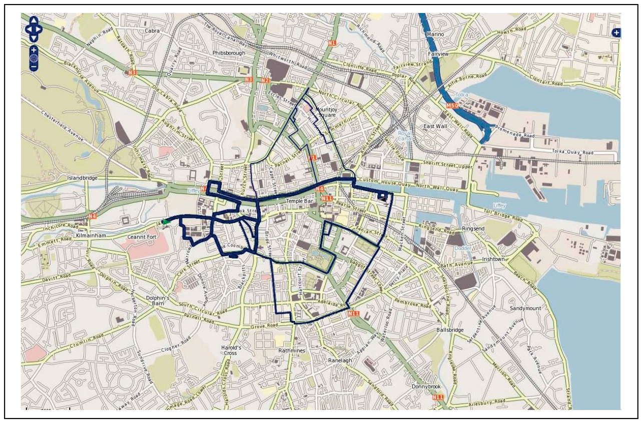

Fig. 7. Three clusters showing distinct routes of User 2 (overlay on map).

both parameters. Based on the statistics and a range of reachability plots we obtained, we found the best combination of values ⇒ *(ϵ = 1000 & minNbs = 3)*. This condition revealed a satisfactory result in terms of the clustering structure from the reachability plots.

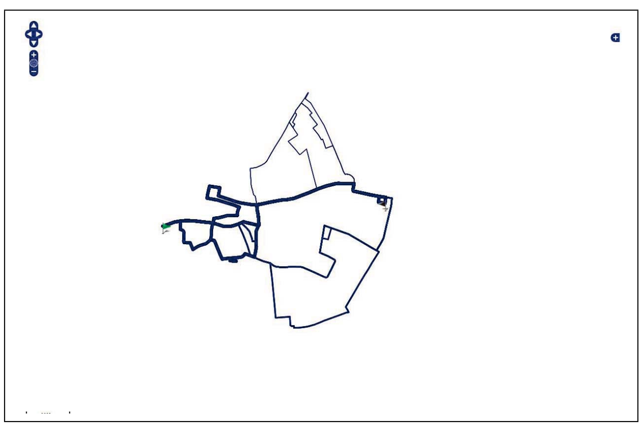

Fig. 8. Three clusters showing distinct routes of User 2 (without overlay).

The reachability plots obtained are illustrated in Figures 4(a) and 4(b). The plots show re-ordering of objects (trajectories in the dataset) on x-axis while y-axis demonstrates the reachability distances between trajectories. Automatic cluster extraction techniques from a graph were presented in [12][22]. This data independent visualization provides analysts a high-level understanding of clustering structure. From these graphs clusters can be identified based on Gaussian-bumps or valleys. As a general rule the cluster starts from a steep-down area and ends at a steep-up area.

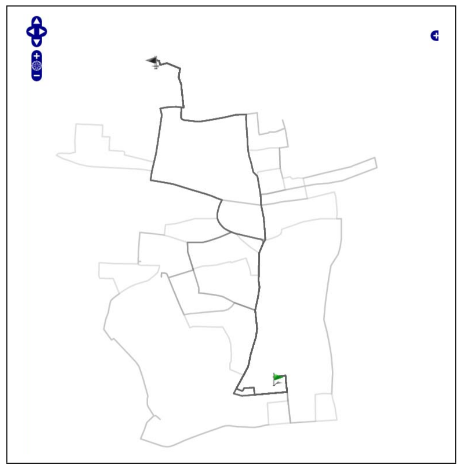

Fig. 9. Best representative aggregated user trajectories (user 1).

Based on the first plot in Figure 4(a), we can clearly see that there are two dominant clusters in user trajectories (trajectory 2 to 13 and trajectory 14 to 25) shown by the valleys in the plot. The other cluster is a group of trajectories, which does not specifically form a valley however they are grouped together into one cluster. The second graph (see Figure 4(b)) also shows three clusters with varying cardinalities (trajectory 2 to 16, 17 to 22 and 23 to 30). In both the graphs, the first trajectory is considered as noise (see OPTICS algorithm [12]). In Figures 5, 6, 7 and 8, the three clusters (from both graphs) are drawn in different styles. The representative routes for each cluster are drawn with different thickness for visualisation purposes.

The clusters show three distinct routes both users adopted over a period of one month to travel from home to work. On average each user trajectory contains almost 100 points. The clustering structure also forms distinct groups based on a specific route on a specific day of the month. For example in Figures 5 and 6, cluster 2 holds trajectories starting from trajectory *14* to trajectory *25* that include *11* days routes. For this specific case we can acquire knowledge about the patterns related with a particular day of a week or a month. For example, if we observe the order in which the trajectories were recorded in case of cluster 2 we obtain *(1,2,3,4,7,8,9,12,13,14,15)*. We can apply heuristics and visualization techniques such as heat maps in order to gain more insights into user behaviors. As apparent from the above sequence user 1 always follows a similar or close

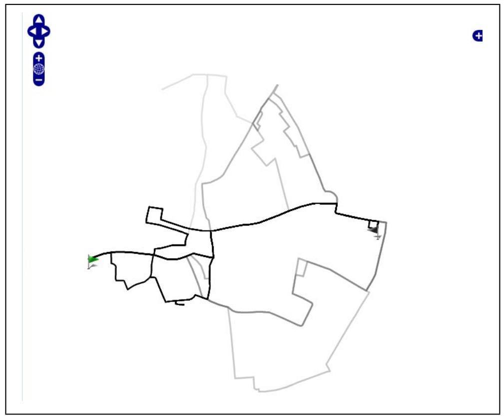

Fig. 10. Best representative aggregated user trajectories (user 2).

route during at least three consecutive days of a month such as *(1,2,3)*, *(7,8,9)* and *(13,14,15)*.

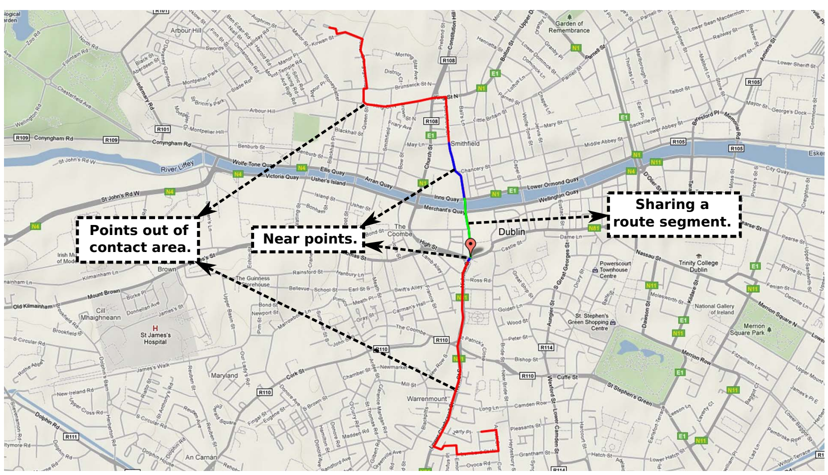

Fig. 11. Best representative trajectory of user 1 in comparison to user 2.

After analyzing the clustering structure the next step is to find an aggregated trajectory or a best representative of a particular user route. For this purpose we have applied a simple yet interesting visualization technique. When all three clusters from both users are visualized using a single grey scale color scheme, it reveals the most frequent route adopted. The color has to be selected in a way that it must be transparent enough to visualize these changes. The phenomenon is illustrated in Figures 9 and 10, where user 1 and user 2 best representatives can be visualized and extracted respectively for further analysis.

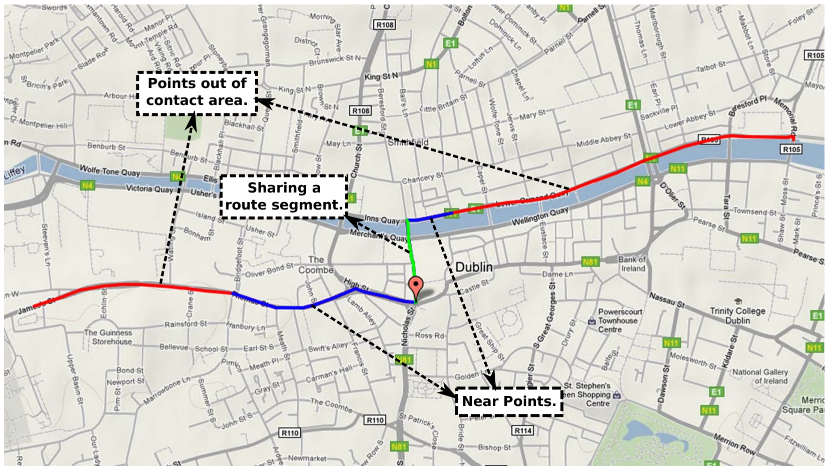

Fig. 12. Best representative trajectory of user 2 in comparison to user 1.

Once the clustering algorithm recognizes the best representative trajectory for each user, the trajectory correlation algorithm is executed. For this example, the algorithm firstly generates the MBRs for each best representative user trajectory and identifies the correlation between both MBRs. After that, it computes the Hausdorff distance of the points in the correlated area.

In order to present the accuracy and efficiency of our system we used a color-based scheme to represent the points in the same road segment, the near points and the points out of the correlated area. Figures 11 and 12 show the trajectory of the users 1 and 2 respectively with the colors representing the near points between them. The green color represents the same segment that is used by both users for their daily routines. The blue color denotes the possible points of interaction, which is in the correlated area among the MBRs. Finally, the red color indicates the points that are out of the correlated area. Additionally, the system allows the generation of messages making use of the context information.

Making use of the presented results and taking into account the use of context information to describe Points of Interests (PoI), we conclude that our approach can be applied to a large number of applications, for instance: to offer a system that increases social interactions in real communities; to develop a system that encourages rides among friends (car pooling).

5 Conclusion and Future Work

Virtual community platforms provide solutions to social connectivity, giving people the capability to share interests, opinions, and personal information with other users. Nevertheless, due to the reduction of social connections in real communities and the absence of context-aware mechanisms in virtual communities, social interactions are frequently missed. As a solution, the users' daily routines, can be captured by mobile social applications and shared in virtual communities in order to improve the social connections in real communities.

This paper presents an approach to explore the capabilities provided by clustering algorithms to analyze user trajectories and extract relevant information from them. We focused on clustering and aggregating multiples trajectories generated by the same user in order to identify habits or preferences. We introduced our trajectory correlation algorithm to find similarities between multiple user trajectories based on each user preference and PoI. Consequently, the near points of interests between two or more users are identified. Taking into account the obtained results, we conclude that our research provided interesting avenues for exploring Location-based Social Network (LBSN) applications.

As future work, we intend to evaluate our algorithm with the MBR expansion process. Besides that, we also aim to use a data-mining algorithm implemented in mobile devices. Therefore, the device allows the trajectory analysis, comparing the current rule with previous rules in order to propose a new personal rule. By using a suitable data-mining algorithm, we can infer the time estimation for a specific segment. Finally, we intend to offer a framework for the development of context-aware systems based on trajectory correlation, focusing on the impact of sharing trajectory information in online social networks as well as their privacy implications [23]. This framework will provide a collection of procedures to acquire, store, increase and infer contextual metadata related to the near points in the correlated area. Additionally, we aim to reuse our techniques in different types of scenarios (for example car pooling and tourism related applications). Finally, in this paper we did not take privacy issues into account; however, these will have to be considered if the application is deployed commercially.

References

[1] Braga, R.B., Martin, H.: Captain: A context-aware system based on personal tracking. In: The 17th International Conference on Distributed Multimedia Systems / DMS 2011, Florence, Italy, KSI (2011)

[2] Wu, Q., Huang, B., Tay, R.: Adaptive path finding for moving objects. In: Web and Wireless Geographical Information Systems. Lecture Notes in Computer Science, Springer (2005) 155–167

[3] Pfoser, D., Brakatsoulas, S., Brosch, P., Umlauft, M., Tryfona, N., Tsironis, G.: Dynamic travel time provision for road networks. In: The 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '08, New York, NY, USA, ACM (2008) 68:1–68:4

[4] Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: driving directions based on taxi trajectories. In: The 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '10, New York, NY, USA, ACM (2010) 99–108

[5] Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., Tennenholtz, M.: Trust-based recommendation systems: an axiomatic approach. In: The 17th International Conference on World Wide Web. WWW '08, New York, NY, USA, ACM (2008) 199–208
[6] Cavanagh, A.: From culture to connection: Internet community studies. Sociology Compass **3** (2009) 1–15
[7] Online Conference on Networks and Communities: Are virtual communities a good thing socially? http://networkconference.netstudies.org. Last access: October 27, 2011 (2010)
[8] Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005)
[9] Morris, B., Trivedi, M.: Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009. (2009) 312 –319
[10] Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., Andrienko, G.: Visually driven analysis of movement data by progressive clustering. Information Visualization **7** (2008) 225–239
[11] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: The 2nd International Conference on Knowledge Discovery and. (1996) 226–231
[12] Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure. SIGMOD Rec. **28** (1999) 49–60
[13] Tahir, G. McArdle, M.B.: Visualising user interaction history to identify web map usage patterns. In: 14th AGILE International Conference on Geographic Information Science, Advancing Geoinformation Science for a Changing World, Utrecht, The Netherlands (2011)
[14] Andrienko, G., Andrienko, N., Wrobel, S.: Visual analytics tools for analysis of movement data. SIGKDD Explorations Newsletter - Special issue on visual analytics **9** (2007) 38–46
[15] Points of Interest Working Group: W3c points of interest working group charter. http://www.w3.org/2010/POI/charter/. Last access: October 27, 2011 (2011)
[16] Benevenuto, F., Rodrigues, T., Cha, M., Almeida, V.: Characterizing user behavior in online social networks. In: The 9th ACM SIGCOMM Conference on Internet Measurement. IMC '09, New York, NY, USA, ACM (2009) 49–62
[17] MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In Cam, L.M.L., Neyman, J., eds.: The 5th Berkeley Symposium on Mathematical Statistics and Probability. Volume 1., University of California Press (1967) 281–297
[18] Papadias, D., Sellis, T., Theodoridis, Y., Egenhofer, M.J.: Topological relations in the world of minimum bounding rectangles: a study with r-trees. ACM SIGMOD International Conference on Management of Data **24** (1995) 92–103
[19] Atallah, M.J.: A linear time algorithm for the hausdorff distance between convex polygons. Informatics Processing Letters **17** (1983) 207–209
[20] Jacox, E.H., Samet, H.: Metric space similarity joins. ACM Transaction on Database Systems **33** (2008) 7:1–7:38
[21] Bischoff, S., Pavic, D., Kobbelt, L.: Automatic restoration of polygon models. ACM Transactions on Graphics **24** (2005) 1332–1352
[22] Brecheisen, S., Kriegel, H., Kröger, P., Pfeifle, M.: Visually mining through cluster hierarchies. In: International Conference on Data Mining, Orlando, FL, Citeseer (2004)

[23] Gross, R., Acquisti, A.: Information revelation and privacy in online social networks. In: Proceedings of the 2005 ACM workshop on Privacy in the electronic society. WPES '05, New York, NY, USA, ACM (2005) 71–80