International Cartographic Association
Association Cartographique Internationale

Real-time destination prediction for mobile users

*Radu Mariescu-Istodor[a], Roxana Ungureanu[b], Pasi Fränti[c]

[a] *University of Eastern Finland, radum@cs.uef.fi*
[b] *University of Eastern Finland, ungureanu.roxana94@gmail.com*
[c] *University of Eastern Finland, franti@cs.uef.fi*
* Corresponding author

Abstract: The number of GPS trajectories recorded daily has been continuously growing in the recent years and new methods to analyse such big data are surfacing all the time. In this paper, we focus on destination prediction, which is useful in various applications like hazard detection and advertisement. We proposed a real-time method for destination prediction of moving users. It uses the current movement trajectory of the user together with historical and regional information to make an accurate prediction. The method is efficient because we can rapidly compute features with the help of spatial and non-spatial indexing methods. We tested the method with real trajectories collected by Mopsi users. The success rate of the method is up to 65 % depending on the length of the recorded trajectory so far, i.e. how long the user has been on move. To our knowledge, this is the first real-time system capable of such success.

Keywords: GPS trajectories, destination prediction, user profiling

1. Introduction

Location-based services are becoming more and more popular, and the market size is said to reach 68.85 billion US dollars by 2023 (Marketsandmarkets, 2018). Mobile users willingly use such services to collect location-based of data for a better user experience. For example, in Flickr, Instagram and Google Photos, users create geo-tagged photo galleries. Other examples are SportsTracker, Endomondo and Strava users who record and analyse their exercise trajectories. Users of Facebook, Google Places and Trip Advisor get personalized ads and location-based recommendations.

Trajectory prediction is a problem that has gotten attention in recent years. It refers to obtaining the trajectory of a mobile user before it happens. There are two categories of application scenarios (Georgiou et al., 2018). The first performs long-term predictions, which are important to optimize objectives such as achieving low cost efficiency or ensuring public transportation. In this work we devised a method to guess the end-point using information about the current movement, the surroundings, and user history. This goal is not easy to achieve mostly because trajectories have varying lengths. In (Krumm, 2010; Krumm, 2016) prediction is limited to only the very near future. These are probability-based methods that use Markov models to determine what happens at the following intersection: does the user turn or continue forward. While these methods work with good accuracy, using them for long-term prediction has the side effect of small errors propagating and leading to a much lower accuracy.

The second approach is to compute short-term prediction with an immediate response. It is useful for warning a driver of upcoming hazards, traffic congestion and collision risk (Ammoun and Nashashibi, 2009). An alternative use is in advertising where on-path services are retrieved using predictive range queries (Jeung et al., 2010) on spatial databases.

Predicting the future trajectory can be done in two steps (Krumm et al., 2013): *destination prediction* and finding the path to the destination. In this paper we focus on the first task and present a method that works in real-time, which, to the best of our knowledge is the first of its kind.

GPS trajectories are sequences of ordered points $T=(p_1,p_2 \ldots p_n)$ and each point typically has two properties: the location (latitude and longitude) and the time stamp. Mobile users typically start recording a trajectory by pressing a Start button (see Figure 1). We will refer to the location at this time as the *start-point*. Then, the app continues to record the trajectory, typically at a fixed interval like every 2-4 seconds. We refer to the most recent location as the *current-point*. The user can stop recording by pressing the Stop button; we refer to the last location as the *end-point*.

Figure 1. Basic tracking features shown in Mopsi app.

In (Krumm et al., 2013), the authors tackle the problem of predicting the destination farther into the future. Their method works by analyzing destination candidates based on the efficiency of travel: does the user appear to move towards the destination candidate or not. There are two problems with this approach. First, processing takes very long, in the order of magnitude of hours, because calculating the driving efficiency to multiple candidates requires many shortest path calculations. Secondly, due to the time concerns, the travel duration is limited to 1 hour into the future according to the expected trip duration inferred from the dataset. This can have an effect on the prediction as trajectory lengths vary significantly depending on the user, the transportation mode, or the street network in the area.

We will investigate different, less computationally expensive features and compare them against this moving efficiency approach in the evaluation section.

Our proposed method is implemented in Mopsi[1]: a location-based social network developed by the School of computing of the University of Eastern Finland. Mopsi has the following main functionalities: geo-tagged photo clustering, trajectory search and analysis, event planning and service recommendation.

2. Method

The method works in two steps (see Figure 2). First, a pre-processing step generates historical profiles for the Mopsi users and indexes them for fast retrieval, which makes it possible to build a real-time system. These profiles include all the past destinations, and movement statistics. These profiles are updated periodically (once a week), to account for changes in behaviour. Second is the actual prediction step, which uses the current movement information, nearby points of interest (POIs) and the user profile information.

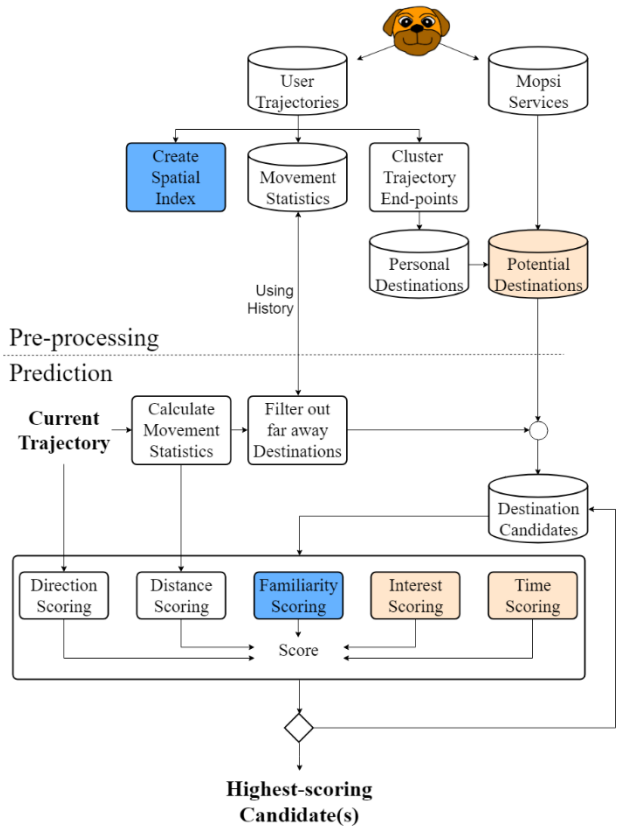

Figure 2. Method workflow.

2.1 Destination Candidates

One source for the potential destinations is the Mopsi services database[2], which is a collection of Points of interest (POIs) such as restaurants, shops and hotels among many others. Other potential source for the destinations are particular to the users, such as their home, work or friend's homes. We generate this set by clustering the end-points of a user's past trajectories using the fast PNN algorithm (Fränti et al., 2000). The resulting cluster centroids are saved into *Personal Destinations* database (see Figure 2). Statistical information like the time and frequency of reaching the clustered destinations is also computed at this stage.

A user may decide to stop anywhere, not necessary at a place in our database. However, most of the time they stop at the calculated destination for reasons we will explain later in the experiments section. The main reason for using a database of meaningful places is that it will significantly reduce the number of tests and, as a result, computation time reduces from hours to seconds. If all road intersections were considered as potential destinations, processing time would be in the order of magnitude of hours (Krumm et al., 2013). A further reason for using

1 http://cs.uef.fi/mopsi

2 http://cs.uef.fi/mopsi/MopsiSet

meaningful locations is that we can utilize also their metadata (keywords) to improve our prediction by matching it to user's personal interests.

Given a current (incomplete) trajectory, we select the nearest places to the user's current-point as destination candidates to be evaluated. This is done as follows. First, automatically infer the transportation mode using the method in (Waga et al., 2012). Knowing this, we look in the historical profile and check the expected distance travelled by the user by the inferred transportation mode and retrieve candidates within this expected distance with a tolerance (see Figure 3). We set a tolerance of two standard deviations and in the unlikely case (5 % chance) that no destination candidates are found, we start to double the radius until some results appear.

2.2 Features

To decide which is the most likely destination among the candidates we score each of them using five features:

1. Time
2. Interest
3. Familiarity
4. Distance
5. Direction

The *Time* feature is used to measure if a given destination candidate is typically being visited at the current time. Considering all past trajectories of a user A and a destination candidate d, we calculate a probability that d is the destination:

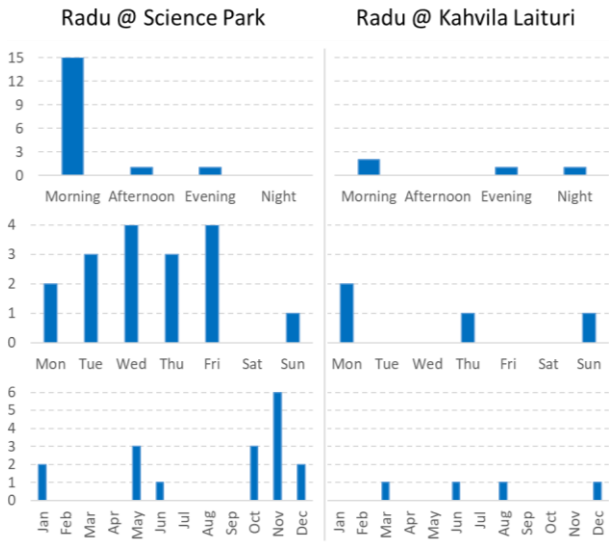

Figure 3. The number of times Mopsi user Radu ended tracking at Science Park and Kahvila Laituri. The data is aggregated by time of day (top), day of week (middle) and month of year (bottom).

$$p(d, A) = \frac{1}{|A|}\sum_{i=1}^{|A|}(\text{end-point}(A_i) = d). \quad (1)$$

We then calculate a time score for d as follows:

$$\text{Time}(d) = \frac{1}{4}(p(d, A) + p(d, T) + p(d, D) + p(d, M)), \quad (2)$$

where A is the set of all trajectories of the user, T is the set of trajectories recorded at the current time of day, D is the set of trajectories recorded on the same day of the week, and M is the set of trajectories recorded in the same month of the year (see example in Figure 3).

The *Interest* feature is similar to the time feature except that the type of the destination candidate is used instead. This feature will provide a useful scoring in places where the user has not visited before. For example, due to the *Kahvila Laituri* visits from Figure 3, all services that serve coffee will increase their score.

The *Familiarity* feature measures if the user has recorded a similar trajectory in the past. If it is a reoccurring pattern, destination candidates near to the end-points of the past similar trajectories receive a higher score (see Figure 4).

$$Familiarity(d) = p(d, S), \quad (3)$$

where S is the set of trajectories which include the current trajectory. To obtain S, we query the Mopsi trajectory database and rank trajectories using the inclusion measure presented in (Mariescu-Istodor and Fränti, 2017), which can be computed in real-time. We then limit to the top ranking candidates by clustering the inclusion values and keeping the top cluster as described in (Mariescu-Istodor and Fränti, 2016).

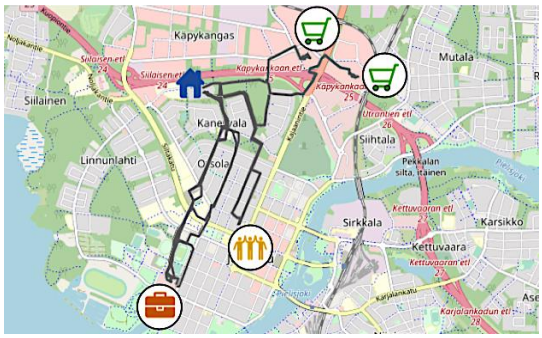

Figure 4. Sample past trajectories leaving Radu's home towards the East and where they end.

The *Distance* feature measures if the destination candidate is within the typical travel distance executed by the user with the given transportation mode. It is measured using a normal distribution with properties inferred from the user profile.

$$Distance(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (4)$$

here μ and σ are the mean travel distance and standard deviation. The intuition for using the transportation mode is that if the user started driving a car, the destination candidate is unlikely to be nearby, whereas if the user is walking, the destination cannot be too far away.

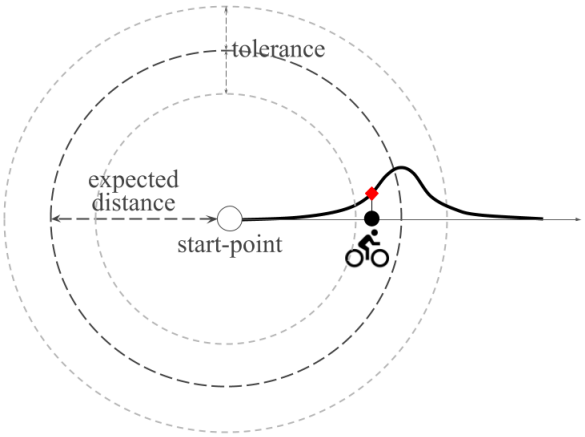

Figure 5. Given a start-point, the user is expected to stop somewhere along a circle with a given tolerance.

Direction is important when users move towards the destination. In this case, we give low scores to candidates in the opposite direction of the user's movement, for instance. The score is calculated as the change in direction (angle) between the current trajectory heading and a given destination candidate. There are situations when a user does not move towards the destination. For example, situations when the movement is restricted by the road network, or, when the user is making a loop (typical when jogging or cycling). We identify these situations by looking at the optimality of the current trajectory. If another, shorter route exists between the start-point and the current-point, it indicates that the movement is not optimal and, therefore, the direction of travel should not be trusted so much, and we therefore scale its significance as follows:

$$\text{Optimality}(T_{1..n}) = \frac{\text{length}(\text{SP}(T_1, T_n))}{\text{length}(T_{1..n})} \quad (5)$$

$$\text{Direction}(T_{1..n}, d) = \text{angle}(T_{n-\alpha}, T_n, d) \cdot \text{Optimality}(T_{1..n}) \quad (6)$$

where SP is the shortest path between the two points. We set a value of $\alpha = 20$, the 20th most recent point.

In literature, the efficiency of travel towards a destination has been used instead of the direction (Krumm and Horvitz, 2006; Krumm et al., 2013). In other words, calculating shortest path to all destination candidates is performed and scoring is based on the relation to the current movement. We experimented with this alternative as well. While it provides more accurate values than the direction feature, the difference was not significant but it increases the computation to 10 times, and therefore, we decided to use our simpler variant for the sake of speed.

3. Evaluation

To evaluate the proposed method we performed experiments on the MopsiRoutes2019[3] trajectory dataset. The dataset contains the most recent trajectories from 10 different Mopsi users recorded by 31.3.2019. These contain a total of 2,484 trajectories consisting of 3,409,812 points, travelling of a total 30,599 km in 2,103 hours. The trajectories were recorded using different means of transport: walking, running, cycling, or by motorized vehicle. They are typically exercise sessions or commuting to stores and work place.

The 100 most recent trajectories of each user were used for testing and all earlier trajectories up to one year (365 days) prior to that were used to generate the profiles.

To evaluate the method, we collected ground truth for all evaluated trajectories. The closest POI to a given trajectory end-point is considered the correct destination. If other POIs are nearby (within 20 meters to the end-point), we accept multiple ground truths in this case. This accounts for errors in GPS accuracy, dataset quality, and multiple entrances in a building.

We first evaluated the usefulness of the features. We tried predicting using a single feature at a time and ranked the results. We noticed the most important feature is the *Familiarity*, followed closely by the *Time* and then the others (see Table 1).

User	Time	Interest	Familiarity	Distance	Direction
Andrei	++		+++	+	
Pasi	++		+++		+
Jukka	++	+	+++		
Karol	+++			++	+
Make	+++			++	+
Zhentian	+++	+		++	
Radu	++		+++	+	
Oili	+++		+++		+++
Matti	++		+++	+	
Yuliya	++		+++		+

Table 1. Usefulness of every feature per user

We then experimented with summing up the features and noticed that the *Interests* score is actually harmful. It improve the prediction for three users but the overall performance is lower. This is likely because these three users were visiting exactly the same services and not some alternatives providing similar service. In the following experiments, we removed this feature from the summation.

3 http://cs.uef.fi/mopsi/routes/2019

One outcome of the experiment is that the method can be evaluated only 74 % of the time. In the rest 26 % of the cases, there were no POIs in the area (nearest POI to end-point was more than 1 km away). This happens when a user is travelling first time to another city or country where no Mopsi services exist.

When the method is applicable, we experiment the effect of how big proportion of the trajectory is known. We also evaluate the outcome in two ways: a hard evaluation where the correct candidate must be the candidate with the highest score, and a soft evaluation when the correct candidate appears in the top three of the prediction results. The results are summarized in Table 2.

There are 222 destination candidates to be evaluated, on average. This means a 0.7 % success rate is expected by a random choice. We notice that the performance is far above that of guessing and it typically increases when more of the trajectory is known; success rate being above 60 % under the soft evaluation. If the entire trajectory (100%) is given as input, the method should ideally terminate and report the nearest candidate. This happens 51 % of the time in the hard evaluation, and 72 % in the soft evaluation. We also measure the distance from the predicted candidate to the end-point of the trajectory and found that, on average, the distance is small - about 20 meters. This suggests that Mopsi services dataset is dense enough.

We compare our results with the method from (Krumm et al., 2013) which we implemented to the best of our ability. To calculate the efficiency of travel, we use the shortest path obtained with Dijkstra's algorithm and contraction hierarchies provided by the *Open Source Routing Machine*[4]. For the road network we use *OpenStreetMap*[5]. We calculated the efficiency score to the candidates using the following formula

$$\text{Efficiency}(T_{1..n}, d) = \frac{\text{SP}(T_0, d)}{\text{length}(T_{1..n}) + \text{SP}(T_n, d)}, \quad (7)$$

where SP is the shortest path between the two points.

In (Krumm et al., 2013), a travel time restriction of 1 hour is imposed. We decided to use our expected travel distance score here instead, and selected the candidate with the highest efficiency score in a 10 % range near to the expected travel distance. This setting provided the best outcome on our data with the results summarized in Table 2.

The efficiency of the travel is not very useful, especially when only little is known of the trajectory. It becomes more useful later; however, the time and familiarity features in our proposed method outperform the efficiency feature by a wide margin. Efficiency is also time consuming to compute. It takes 22.5 seconds per trajectory, on average, compared to our proposed approach which performs the prediction in real-time (1.2 seconds on average).

	Proposed method		**Compared method**	
Trajectory Proportion	Hard evaluation	Soft evaluation	Hard evaluation	Soft evaluation
25 %	37 %	55 %	2.5 %	9.5 %
50 %	45 %	62 %	5.7 %	9.9 %
75 %	49 %	65 %	11 %	18 %
100 %	51 %	72 %	28 %	38 %

Table 2. Performance of the compared methods when different proportion of the trajectory is known. Hard and soft evaluation is done for each.

We next simulate a cold-start situation by generating an average profile by combining the history of multiple users and testing its predicting ability on other users' trajectories. We consider the same 10 users as before to create the average profile. However, when testing on a specific user, the profile is altered by removing his historical data for a fair evaluation.

	Personal profile		**Average profile**	
User	Hard evaluation	Soft evaluation	Hard evaluation	Soft evaluation
Andrei	**51.0 %**	**64.0 %**	4.0 %	9.0 %
Pasi	**43.0 %**	**64.0 %**	1.0 %	1.0 %
Jukka	37.0 %	52.0 %	**5.0 %**	**16.0 %**
Karol	29.0 %	51.0 %	**15.0 %**	**34.0 %**
Make	49.0 %	54.0 %	**12.0 %**	**15.0 %**
Zhentian	40.0 %	48.0 %	**22.0 %**	**30.0 %**
Radu	39.0 %	53.0 %	0.0 %	0.0 %
Oili	**40.0 %**	**80.0 %**	0.0 %	0.0 %
Matti	**78.0 %**	**86.0 %**	**20.0 %**	**30.0 %**
Yuliya	**47.0 %**	**65.0 %**	1.0 %	2.0 %
TOTAL	**45.30 %**	**61.70 %**	**8.00 %**	**13.70 %**

Table 3. Performance of the methods using personal and average profiles per user when trajectory proportion is set to 50%. The best results for each profile are shown in **blue**.

Table 3 shows the performance of this model compared with the personal model when 50% of each trajectory is known. The performance is significantly lower, but still

[4] http://project-osrm.org

[5] https://www.openstreetmap.org

better than the method provided by (Krumm et al., 2013). It also appears that for half of the users the performance is very poor (hard = 1.2%, soft = 2.4%) while for the better half it is quite high in comparison (hard = 14.8%, soft = 25%). This low performance can be explained by the fact that users like Andrei, Pasi and Radu often end tracking at their homes (a personal POI). Removing this information from the average profile has severe consequences. Also users Oili and Yuliya are poorly predicted but here the reason is that they travel to locations that other people don't: travelling abroad in various countries and to local gym or own home, respectively. Other users for which the system provides better prediction have common behaviours like living in the same location or shopping in the same place.

Not all users record the same number of trajectories. This amount varies significantly as seen in Figure 6. Therefore, the one year history we used is more detailed for some users than others. There is a surprising negative correlation (Pearsons = -0.25) between the number of trajectories and the predicting ability. One would expect that a more complete profile would make for a better prediction, however, the situation here is that users that record less tracks have, in general, more predictable behavior such as home-to-work and home-to-shop trajectories, while users that have more data tend to have more types of activities such as cycling, skiing, driving, and orienteering.

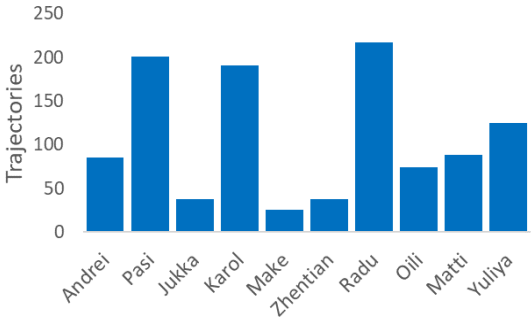

Figure 6. Number of trajectories recorded per year by the 10 Mopsi users.

In Figure 7, we see a real example using the tools available in Mopsi: a home to work trajectory of the user Radu. The time feature alone suggests that the end-point would be his home, and the workplace is ranked as the second choice. This is because the user typically jogs around this time and returns home afterwards.

Interest feature indicates multiple candidates with dining possibilities as the user often eats breakfast around the same time. Familiarity feature indicate endpoints that contain the traveled section; majority of them end at home again. Distance feature provides relatively high score because all locations are within reach. Scores are slightly lower close to home and higher towards city center but the difference is not very noticeable in the interface. Direction feature provides scores that are higher in the direction of the travel. When all the scores are summed up, the correct destination is detected.

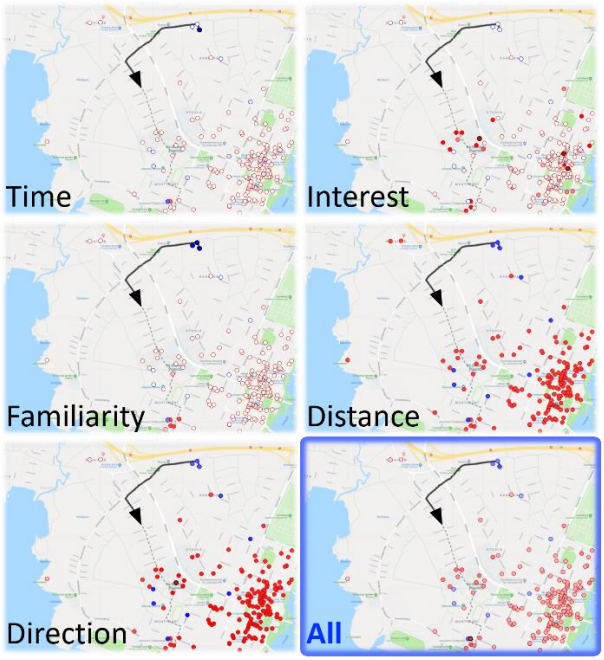

Figure 7. Example trajectory and scoring for each feature individually and combined. The intensity of the color is proportional to the score. Red dots are Mopsi services, blue dots are personal POIs. The dotted line is the unseen part of the trajectory.

More qualitative examples are shown in Figure 8. Example A is a long running track which is a loop and the correct destination is correctly detected as the starting point. Example B shows returning home from visiting a friend. Example C is an example when method fails. It predicts the location where the user used to live. User moved to a new home recently. Finally, example D is a team building cycling trip organized with colleagues and students at work. Destination is back to work again.

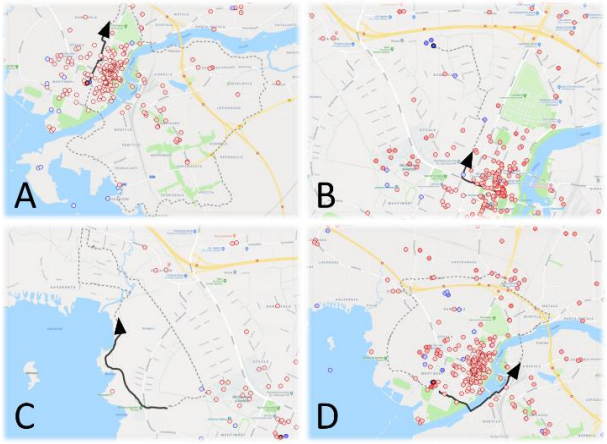

Figure 8. Qualitative examples.

4. Discussion

The proposed method presents a clear advantage over other similar systems is its real-time ability to make the prediction and still achieve high accuracy. While we demonstrate that the method works well, the number of users used for testing is quite low and a more solid evaluation should be performed to confirm the results. This can be possible if more data would be available in the future.

We demonstrated that the method can also work with cold-start users. However, the system must have data from other users in the database. Otherwise, an average profile cannot be computed. Another drawback is that if services (POIs) are not available in a region of interest, the method cannot work.

5. Conclusions

We proposed a method to predict the destination of a mobile user when knowing part of the current trajectory, past movement history and nearby points of interest. The method was demonstrated to work in real-time with up to 65 % success rate using real data collected by Mopsi users.

The most important features are the familiarity and the time, followed by the direction, distance and interest. However, this is user specific and experiments suggest that a higher performance may be achieved by using machine learning to provide better weights for the feature values. This qualifies as a future work.

In 26 % of the test cases, the method was not applicable because of limited dataset of POIs. However, this problem can be removed by using a more complete dataset instead of the Mopsi dataset.

As a final remark, our study indicates that security may be of concern for people who track their movement, especially if they have a predictable life-style. This concern relates mostly to public data or if a system was hacked.

6. References

Ammoun, S. and Nashashibi, F. (2009). Real time trajectory prediction for collision risk estimation between vehicles. IEEE 5th International Conference on Intelligent Computer Communication and Processing, pp. 417-422.

Fränti, P., Kaukoranta, T., Shen, D.-F. and Chang, K.-S., (2000). Fast and memory efficient implementation of the exact PNN. IEEE Trans. on Image Processing, 9 (5), 773-777

Georgiou, H., Karagiorgou, S., Kontoulis, Y., Pelekis, N., Petrou, P., Scarlatti, D. and Theodoridis, Y. (2018). Moving Objects Analytics: Survey on Future Location & Trajectory Prediction Methods, arXiv preprint arXiv:1807.04639.

Jeung, H., Yiu, M.L., Zhou, X. and Jensen, C.S. Jensen. (2010). Path prediction and predictive range querying in road network databases. The VLDB Journal 19, no. 4 : 585-602.

Krumm, J. and Horvitz, E. (2006). Predestination: Inferring destinations from partial trajectories. In International Conference on Ubiquitous Computing (pp. 243-260). Springer, Berlin, Heidelberg.

Krumm, J. (2010). Where will they turn: predicting turn proportions at intersections. Personal and Ubiquitous Computing 14, no. 7: 591-599.

Krumm, J., Gruen, R. and Delling, D. (2013). From destination prediction to route prediction. Journal of Location Based Services 7, no. 2: 98-120.

Krumm, J. (2016). A markov model for driver turn prediction.

Marketsandmarkets. (2018). Location-Based Services (LBS) and Real-Time Location Systems (RTLS) Market by Location Type, Software, 537 Hardware, Service, Vertical, and Region - Global Forecast to 2023; p. 183.

Mariescu-Istodor, R. and Fränti, P. (2017). Grid-based method for GPS route analysis for retrieval. ACM Transactions on Spatial Algorithms and Systems (TSAS), 3(3), 8.

Mariescu-Istodor, R. and Fränti, P. (2016). Gesture Input for GPS Route Search. In Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR) (pp. 439-449). Springer, Cham.

Waga, K., Tabarcea, A., Chen, M. and Fränti, P. (2012). Detecting movement type by route segmentation and classification. In 8th IEEE International Conference on Collaborative computing: networking, applications and worksharing (CollaborateCom) (pp. 508-513).

Zheng, Y., Zhang, L., Xie, X. and Ma, W-Y. (2009) Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of the 18th ACM international conference on World wide web, pp. 791-800.