Detecting Location-Based User Actions

Author(s):
Mariescu-Istodor, Radu; Fränti, Pasi

Publication Date:
2018-01-15

Permanent Link:
https://doi.org/10.3929/ethz-b-000225579 →

Detecting Location-Based User Actions

Radu Mariescu-Istodor, Pasi Fränti

Machine Learning Group, School of Computing
University of Eastern Finland, Joensuu, Finland
Email: {radum,franti}@cs.uef.fi

Abstract. The use of mobile devices with GPS facility is increasing in daily life. In this paper, we propose to detect several *user actions* based on data collected by the Mopsi[1] application. We distinguish several types of actions such as: taking or uploading photos, completing tracking, visiting or passing-by a service and multiple users meeting. We evaluate the proposed algorithms by collecting ground truth and measuring error rates. The proposed methods perform with an average error of 5.9% and a weighted average error (WA) of 2.8%.

Keywords. user action, location-aware, clustering

1. Introduction

Computing devices have become ubiquitous and increasingly mobile in the past decade[2]. GPS sensors are nowadays present in 85% of mobile devices, which means that it is now common for apps to access a user's geographical position. User actions are statements describing a user's behavior at a certain time. Detecting location-based actions is important in various situations. For example, it helps social interaction by keeping users aware of how, when and where their friends interact.

In (Xiao et al. 2012), the authors describe a method for inferring user similarity based on similar actions. They find places where users visit by first detecting stopping points in their trajectories and then map the stop to service locations from a separate database. This visit detection system works

[1] cs.uef.fi/mopsi

[2] businessinsider.com/15-billion-smartphones-in-the-world-22013-2

for data collected over a period of time because it relies on finding significantly long stops in user movement. We, however, aim to detect the actions in real-time. We demonstrate our methods in Mopsi, a location-based social network developed by the Machine Learning Unit, School of Computing in the University of Eastern Finland. Mopsi provides services such as search and recommendation of services, data collection and user tracking. The contribution of this paper is to propose new algorithms for detecting user actions and to evaluate them by collecting ground truth from users.

2. User Actions

We distinguish several types of actions such as: taking or uploading photos, completing tracking, visiting, passing-by or leaving a place and multiple users meeting (Mariescu-Istodor 2013). We classify user actions in three categories based on how the detection is made: triggered actions, analyzed actions and triggered actions with additional analysis. In the following subsections we elaborate on each class in more details.

2.1. Triggered Actions

These actions are recognized upon a single HTTP request made to the server by the mobile device of a user. They do not require additional system information in order to be detected.

Taking or uploading photo: We differentiate two separate actions when a photo reaches the server: photo taking and photo uploading. If a photo reaches the server soon after it was taken (under 10 minutes) the *photo taking* action is triggered; otherwise, we trigger the *photo uploading* action.

2.2. Analyzed Actions

These actions are based on analysis of collected data. This analysis is performed periodically, every 30 seconds.

3.2.1. Visiting Places

The places we consider are the Mopsi services: restaurants, bars, shops, etc. Three action types can be concluded when a user comes in the vicinity of a place:

- *Visit*: staying at a place for a considerable amount of time;
- *Leave*: moving away from a place they were visiting;
- *Pass-by*: the user is near the service but not visiting.

Detecting these scenarios is not trivial due to inaccuracy in GPS signal, which can make it look like a user is moving away from a place even though user is standing still. We solve this by introducing *link method*. A link is a virtual connection established between a user and the nearest service when the user is in ε proximity. We associate a *strength* value (from 0 to 10) to the link. When the link is established, the strength is initialized to 5 allowing it to update in both directions. In the course of time, the strength value increases if the service remains nearest to the user and decreases otherwise.

If the value reaches 0, the *pass-by* action is detected. A *visit* is detected when the link strength reaches 10. This corresponds to the user staying at the location of the service for a considerable amount of time (typically around 3 minutes, depending on user movement and GPS accuracy). When the value reaches 0 after a visit was detected, a *leave* action is concluded.

3.2.2. Meetings

We define a *meeting* as a situation when multiple users are in close proximity to each other, in other words, part of a group. To find these groups, we cluster the user locations. We use single-link agglomerative clustering (Jain & Dubes 1988) and apply a distance threshold, ε, under which the clusters will merge. Single-link clustering requires $O(n^3)$ where n is the number of online users. Algorithms with better time complexity exist for mean square error criterion. In (Fränti et al. 2006), the authors use a k-nearest neighbor graph to reduce the number of distance calculations iteratively with a time complexity of $O(\tau n \log n)$, where τ is the number of nearest neighbor updates per iteration. This solution will work even if users create a pattern such as people cycling on a narrow street (see Figure 2). The resulting connected components (Leiserson et al. 2001) are groups of users together at a moment in time.

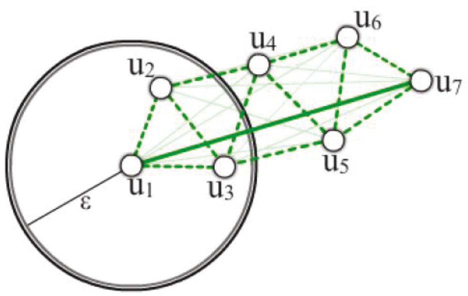

Figure 2. A group of users u_1, ..., u_7 are connected by links of maximum ε meters.

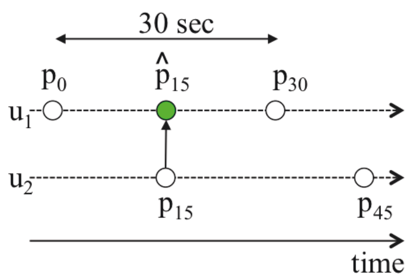

Figure 3. Two users' successive locations. We find $\hat{p}_{15}$ using interpolation.

The method described so far does not work when users are moving together in a car, train or by walking. In Figure 3 two users are moving together at a constant speed, sending location updates every 30 seconds. The users may not send the data at the same time and it appears that they are not together.

We handle this problem by using interpolation to approximate all user locations at a common time. Using the user groups resulting from clustering, we apply the link method to detect actual meetings afterwards.

2.3. Triggered actions with additional analysis

Tracking refers to recording a sequence of points at a fixed time interval. In Mopsi the interval is usually 1, 2 or 4 seconds. Lower frequency is not really needed because the route reduction procedure (Chen et al. 2012) can handle the large amount of data while keeping the accuracy in a given threshold. In an ideal system any new point is immediately sent to the server. In Mopsi, however, we use buffering because network connection may become unavailable without warning. This ensures that no points are lost during the upload process. Upon the completion of the route, movement type analysis is made on the server. The movement type(s) of a route are presented in Mopsi (Waga et al. 2012).

3. Experimental Results

We evaluate the user action detection system using real information collected from Mopsi users during August 2013. The nine most active users were willing to participate in a survey[3]. Each user was asked to mark a checkbox for each correctly detected action. If the action did not happen users were asked to specify the reason in a comment field. If a user did not remember, she was asked to leave the checkbox unchecked and the comment field empty. In this way, we discard the uncertain actions from evaluation.

A total of 1197 actions were detected for the 9 users, see Table 1. Location of the actions varied significantly as the users traveled in six countries: Finland, Romania, Germany, Latvia, Lithuania and the United Kingdom. The most common actions detected were the photo taking or uploading. These are followed by the meeting, tracking, pass-by, visit and leave actions in this particular order. The visit and leave actions are the fewest.

We calculate an unweighted average error (UA) and a weighted average error (WA) for the action detection. The motivation of employing weighted and unweighted average errors is because of the uneven distribution of performed actions. By analyzing the data provided in Table 1 and by the outcome of the surveys, we have UA=5.9% and WA=2.8%. The reason for WA being less than UA is that more frequent actions have less detection errors and the weighted accuracy is dominated by the actions of high frequency. In

[3] sample survey: http://cs.uef.fi/~radum/actionsGroundTruth/pasi

Mopsi, photo actions represent 75% of all detected actions. The weight of these actions combined with a low error in detection (0.1%) decrease the WA. Figure 4 summarizes the errors for the nine users that participated in the survey.

	Action (a)						
User (u)	Photo	Meeting	Tracking	Pass-by	Visit	Leave	Total
Karol	7	5	5	10	0	0	27
Mikko	2	0	3	0	0	0	5
Mohamed	27	18	1	16	3	3	68
Oili	434	3	2	2	0	0	441
Mohammad	4	2	5	0	0	0	11
Pasi	247	30	27	13	2	2	321
Radu	17	3	2	0	0	0	22
Andrei	40	29	22	10	2	2	105
Julinka	125	14	30	26	1	1	197
Total	903	104	97	77	8	8	**1197**

Table 1. Number of detections for each action type per user.

Taking and uploading photo are triggered actions and unlikely to fail: error of 0.1%. Completing tracking is also a *triggered* action but the analysis of the route can produce incorrect movement type. In particular, current algorithm cannot differentiate accurately between fast running and cycling, or fast cycling and traveling by car.

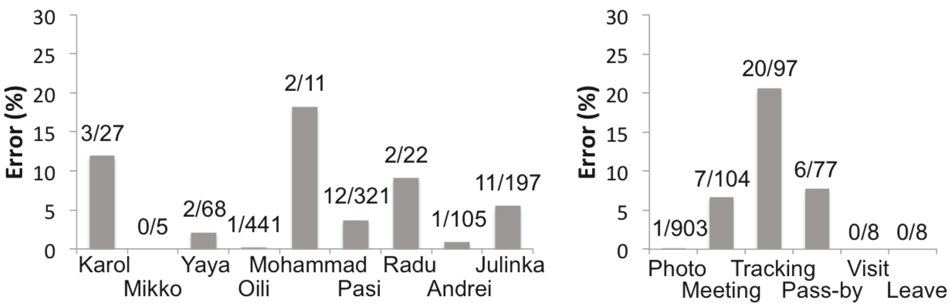

Figure 4. Analysis of Error, by user (left) and by action type (right).

Meetings are detected with an error of 6.7% as indicated in Figure 5. Comments from the users about false meeting detection show two typical failures: consecutive detections instead of one that lasted longer, and users being close to each other but not meeting.

The pass-by misdetections are mostly caused by the fact that when the GPS sensor starts, it takes time before location is updated, and the previous known location was used until then. This behavior may result in showing a user near a service in a wrong time. The visit and leave actions are not as sensitive because they imply user staying at the location for a longer time.

We note, however, that these actions are few compared to others and more testing is needed in this area.

4. Conclusions

We presented methods for detecting user actions and performed a system evaluation by collecting ground truth and calculating weighted and unweighted average error rates. These results measure the correctness of actions but not what was missing. This is because it is hard for users to remember afterwards what actions they did if they are not pointed out by the survey. This study opens up avenues in location-based research. Location clusters may be used in order to acknowledge relevant places that are not explicitly in our database: for example a user's home and work place. User similarity can be evaluated by looking at the actions they have in common. This will benefit recommendation systems and provide a basis for proposing friend connections.

References

Chen M, Xu M, Fränti P (2012) A Fast O(N) Multiresolution Polygonal Approximation Algorithm for GPS Trajectory Simplification. IEEE Transactions on Image Processing, vol. 21, pp. 2770–2785.

Fränti P, Chen C, Tabarcea A (2011) Four aspects of relevance in sharing location-based media: Content, time, location and network. International Conference on Web Information Systems and Technologies (WEBIST), pp. 413–417.

Fränti P, Virmajoki O, Hautamäki V (2006) Fast agglomerative clustering using a k-nearest neighbor graph. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pp. 1875–1881.

Jain A K, Dubes R C (1988) Algorithms for clustering data, Prentice-Hall.

Leiserson C E, Rivest R L, Stein C, Cormen T H (2001), Introduction to algorithms. The MIT press.

Mariescu-Istodor R (2013) Detecting user actions in MOPSI. M.S. thesis, University of Eastern Finland.

Waga K, Tabarcea A, Chen M, Fränti P (2012) Detecting movement type by route segmentation and classification. 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 508–513.

Xiao X, Zheng Y, Luo Q, Xie X (2012) Inferring social ties between users with human location history. Journal of Ambient Intelligence and Humanized Computing, pp. 1–17.