Extracting Meaningful User Locations from Temporally Annotated Geospatial Data

Alasdair Thomason(✉), Nathan Griffiths, and Matthew Leeke

University of Warwick, Coventry CV4 7AL, UK
{ali,nathan,matt}@dcs.warwick.ac.uk

Abstract. The pervasive nature of location-aware devices has enabled the collection of geospatial data for the provision of personalised services. Despite this, the extraction of meaningful user locations from temporally annotated geospatial data remains an open problem. Meaningful location extraction is typically considered to be a 2-step process, consisting of visit extraction and clustering. This paper evaluates techniques for meaningful location extraction, with an emphasis on visit extraction. In particular, we propose an algorithm for the extraction of visits that does not impose a minimum bound on visit duration and makes no assumption of evenly spaced observation.

Keywords: Clustering · Extraction · Geospatial · Location · Visits

1 Introduction

To leverage location-aware devices for service enhancement, systems must be able to interpret and reason about the movements of users. The extraction of meaningful locations from temporally annotated location data is central to achieving this goal, permitting the labelling of locations, e.g., 'home', 'work' or 'supermarket', which increases the capacity of systems to reason about user locations. In particular, meaningful location extraction is fundamental to location prediction, since it establishes the grouping of disparate but related sensor readings.

As vital as meaningful location extraction is to location-based services, the problem remains open. This paper proposes a novel algorithm for *visit extraction*, which is the first stage of meaningful location extraction. The proposed algorithm builds upon previous work and does not impose a minimum bound on visit duration, or have an assumption of evenly spaced location observations. We then evaluate the performance difference between the proposed algorithm in meaningful location extraction and the STA visit extractor [2].

2 Related Work

Extracting locations from a geospatial dataset is often considered a clustering problem. However, when the dataset from which meaningful locations are to be

R. Giaffreda et al. (Eds.): IoT360 2014, Part II, LNICST 151, pp. 84–90, 2015.
DOI: 10.1007/978-3-319-19743-2_13

extracted is temporally annotated, this additional information can be leveraged. *Visit extraction* is concerned with detecting periods of time during which a user remained at a single location, henceforth referred to as *visits*, and consequently summarising the dataset. Traditionally, a road travelled frequently by a user would contain many points, while a location visited only once would contain few points, leading to the possibility of it being overlooked. This summary reduces the computational cost of clustering, as the size of the dataset is reduced.

A discussion of existing approaches to visit extraction can be found in [2]. Such approaches include the use of thresholds to specify maximum visit size relative to either the first point discovered [5,10] or to the visit's centroid [3], but this is highly sensitive to noisy data. Other approaches have used knowledge of the properties of specific GPS devices [1], but such methods are not generic.

Addressing these issues, Bamis and Savvides presented their algorithm for the extraction of *Spatio-Temporal Activities (STAs)* [2]. Although aiming to determine activities that repeat in cycles, they first extract periods of time spent at a single location. The algorithm uses a filter and a buffer of points to detect a consistent change in location of the user, and hence, when the user ends an activity. In contrast to previous approaches, this method is far more resilient to noise. However, the algorithm assumes that points will arrive at even time intervals, and it requires that the buffer be full before a visit can exist, in turn requiring the user to know the minimum length of a visit a priori. The algorithm presented in this paper, *GVE*, does not have such requirements.

Techniques shown to be applicable to the clustering of extracted visits into meaningful locations include k-means [1,7] and DBSCAN [4,6,8,9]. DBSCAN is more popular for this domain as it does not require the number of locations to be known a priori, and is therefore the algorithm adopted in this paper.

3 Gradient-Based Visit Extractor

We propose a Gradient-based Visit Extractor (GVE, Algorithm 1) which extracts visits from temporally annotated geospatial datasets, addressing some of the drawbacks in the STA visit extractor proposed in [2]. GVE works linearly over the dataset by building visits until adding another point would cause the recent trend of motion to be consistently away from the visit already extracted. Although similar in idea to STA, GVE can consider visits without having a full buffer of points over which to analyse the trend of motion and allows for points collected at a varying rate.

The buffer over which the trend of motion of the user is considered has a maximum size of N_{points}, but the buffer does not need to be filled for a comparison to take place. Parameters, α and β, are used to define a threshold function on the size of the buffer. If the buffer contains a small number of points, adding an additional point that is further from p_1 than p_2 could be an indication that the user is moving away from the visit or it could be attributed to noise. This problem is combated by using a negative logarithmic function to ensure that the threshold for trend of motion is higher with fewer points in the buffer. Trend of motion is

Algorithm 1. Gradient-based Visit Extractor Algorithm

```
1: N_points, α, β ← input parameters
2: visits ← [ ] empty array, to be filled with visits
3: visit ← [ p_0 ] array containing the first point in the dataset
4:
5: function Process(point)
6:     if MovingAway?(visit, point) then
7:         visits.append(visit) if visit.length > 1
8:         visit ← [ point ]
9:     else
10:        visit.append(point)
11:    end if
12: end function
13:
14: function MovingAway?(visit, point)
15:    buffer ← visit.last(N_points − 1) + point
16:    return Gradient(buffer, visit) > Threshold(buffer.length)
17: end function
```

defined using a gradient, that includes both spatial and temporal components and therefore allows for the possibility of points of varying temporal distances. The *gradient* of the buffer is defined as:

$$Gradient(b) = \frac{l(b)\sum_{p\in b}(t(p)\times d(p)) - \sum_{p\in b} t(p)\sum_{p\in b} d(p)}{l(b)\sum_{p\in b} t(p)^2 - (\sum_{p\in b} t(p))^2}$$

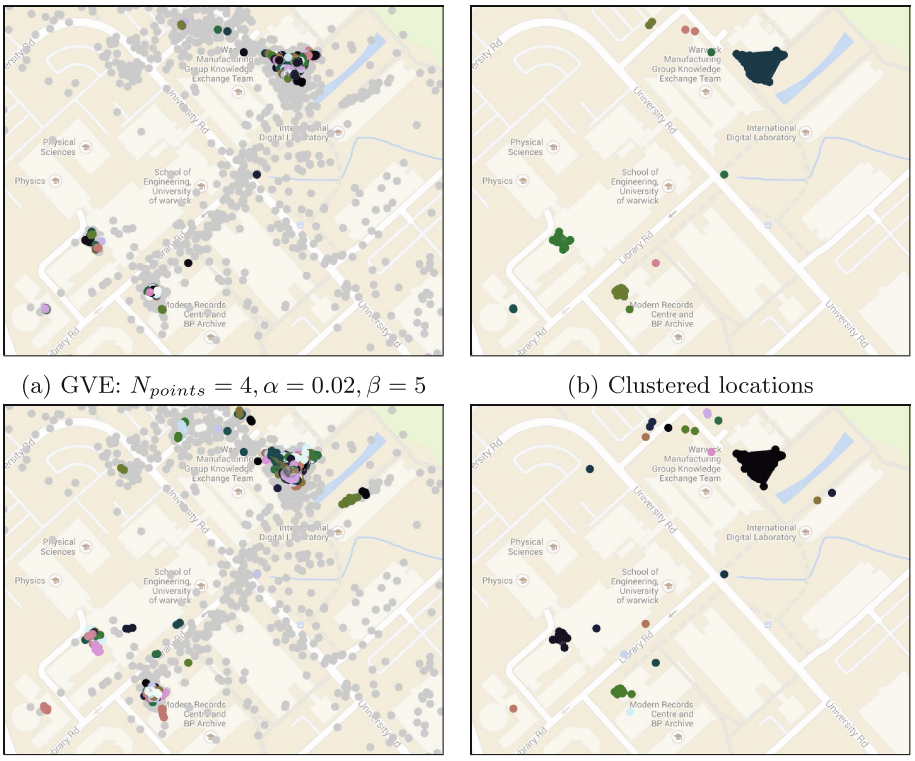
where $l(b)$ is the length of buffer b, $t(p)$ is the time since the first point of the buffer for point p in seconds, and $d(p)$ is the distance between point p and the centroid of the current visit, in metres. A gradient greater than the threshold indicates that the visit has ended:

$$Threshold(length) = -log\left(length * \frac{1}{\beta}\right) * \alpha$$

By combining these two equations, we are able to summarise the movement trend of the user relative to the visit, the *gradient*, and set a threshold for this gradient dependent upon the number of points that it was drawn over. This ensures resilience to noise by monitoring the movement trend over a set of points, but still allows for visits with few points.

4 Evaluation

Using data collected over several months from a smartphone application and a map of the University of Warwick campus, the parameters for the two algorithms were empirically determined such that the locations extracted were consistent with expectations. Results are presented for GVE (Fig. 1) and the STA



(a) GVE: $N_{points} = 4, \alpha = 0.02, \beta = 5$ (b) Clustered locations

(c) GVE: $N_{points} = 10, \alpha = 0.04, \beta = 11$ (d) Clustered locations

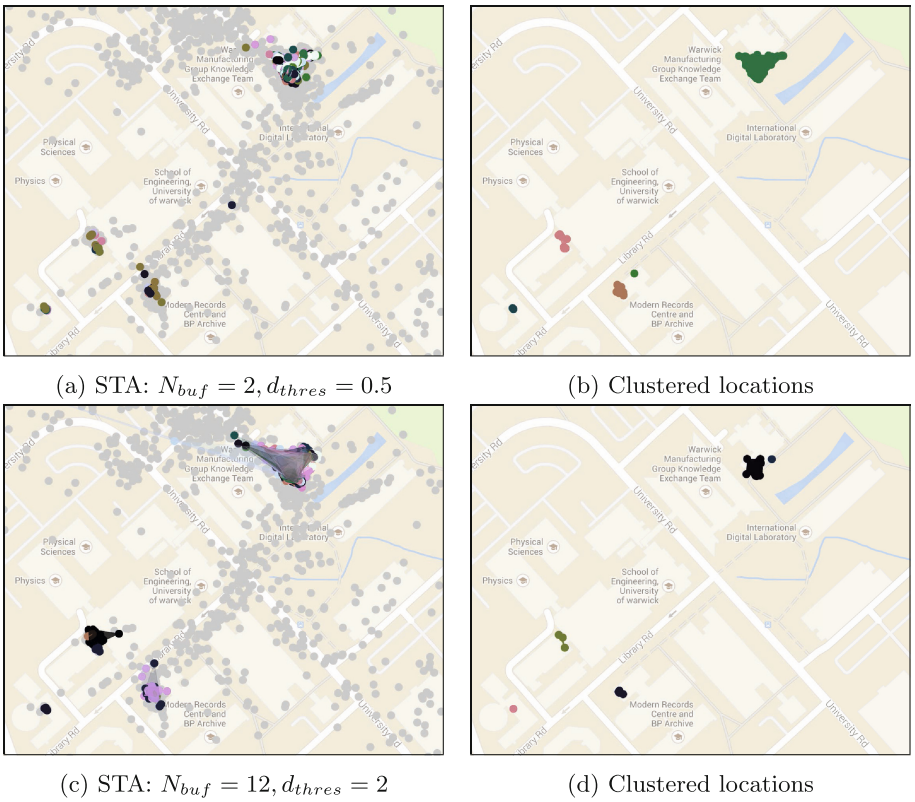
Fig. 1. Gradient-based Visit Extractor

visit extractor (Fig. 2), where the visits identified are clustered using DBSCAN. Results for each algorithm are presented for parameters optimised for accuracy of extracted location and coverage of visits. An immediate observation is that a similar set of primary locations is extracted in all cases, with smaller locations being extracted in only some cases. Specifically, GVE extracts several more locations than STA, since GVE is likely to extract visits of shorter duration. This is substantiated by results in Table 1 where the properties of the extracted visits and locations are detailed. It can be seen that GVE routinely extracts visits of shorter duration, extracting visits of 1 min. STA is capable of extracting similar length visits (1.6 min), but requires that the buffer size be reduced to its minimum of 2. With a larger buffer size (N_{buf} parameter), the minimum length visit extracted is 12.7 min. From the results, it can be seen that when STA is tuned to allow the extraction of short visits, the average and maximum visit lengths are reduced, whilst the total time covered by visits is also reduced.

The algorithm and parameters that produces the greatest temporal coverage is the second run of GVE (with buffer size, N_{points}, of 10). 7.3 days of visits are extracted from the dataset, significantly higher than any other run. This increase

Table 1. Summary of visit extractor results

	α	β	Buffer	D_{thres}	N_d	Visits					loc
						count	avg	min	max	total	
GVE	0.02	5	4			1360	3.4 min	1.0 min	1.5 hr	3.1 days	10
GVE	0.04	11	10			624	16.7 min	1.0 min	3.4 hr	7.3 days	20
STA			2	0.5	1	828	4.8 min	1.6 min	1.8 hr	2.8 days	5
STA			12	2	6	83	2.0 hr	12.7 min	7.7 hr	4.6 days	5



(a) STA: $N_{buf} = 2, d_{thres} = 0.5$

(b) Clustered locations

(c) STA: $N_{buf} = 12, d_{thres} = 2$

(d) Clustered locations

Fig. 2. Spatio-temporal activity extractor

in coverage produces a larger set of locations (as shown in Fig. 3d). Interestingly, however, the number of visits is reduced from the previous run of GVE. This finding indicates that visits which were being split into multiple parts using a smaller buffer size were detected as single visits when a larger buffer is used. Figure 1c shows the visits extracted as representative, with no visit clearly spanning multiple buildings, indicating that the accuracy of extraction has not been impacted.

(a) STA, $N_{buf} = 2$ (b) STA, $N_{buf} = 8$ (c) GVE, $N_{points} = 8$

Fig. 3. GVE and STA Extractor on a visit with few points

An example of the difference between the two algorithms for visits of short duration can be seen in Fig. 3. In the data there exists one visit to the library (bottom right) and two visits to the biology concourse (top left). While it is possible for STA to extract the library visit, the buffer size must be set to 2 which comes at the cost of extracting erroneous biology concourse visits. Selecting parameters that optimise the extraction of visits to the biology concourse means that the algorithm is no longer capable of extracting the short visit to the library (Fig. 3b). Figure 3c shows the results of using GVE to extract the visits. In this case, GVE is capable of extracting all 3 visits correctly.

5 Conclusion

This paper explored the use of visit extraction to better extract meaningful locations from temporally-annotated geospatial datasets. Specifically, a novel algorithm, GVE, has been presented. The algorithm builds on existing work but removed the requirements for visits to have a minimum duration and the dataset to contain points at a constant rate. Further, the paper demonstrated the workings of the GVE algorithm and how it relates to STA for the purpose of extracting visits to aid in meaningful location extraction.

References

1. Ashbrook, D., Starner, T.: Learning significant locations and predicting user movement with GPS. In: ISWC, pp. 101–108 (2002)
2. Bamis, A., Savvides, A.: Lightweight extraction of frequent spatio-temporal activities from GPS traces. In: RTSS, pp. 281–291 (2010)
3. Kang, J.H., Welbourne, W., Stewart, B.: Extracting places from traces of locations. In: WMASH, pp. 110–118 (2004)
4. Lei, P., Shen, T., Peng, W., Su, I.: Exploring spatial-temporal trajectory model for location prediction. In: MDM, pp. 58–67 (2011)
5. Liu, S., Cao, H., Li, L., Zhou, M.: Predicting stay time of mobile users with contextual information. IEEE T-ASE **10**(4), 1026–1036 (2013)
6. Mamoulis, N., Cao, H., Kollios, G.: Mining, indexing, and querying historical spatiotemporal data. In: KDD, pp. 236–245 (2004)
7. Nguyen, L., Cheng, H.T., Wu, P., Buthpitiya, S.: Pnlum: system for prediction of next location for users with mobility. In: Mobile Data Challenge at Pervasive (2012)

8. Palma, A.T., Bogorny, V., Kuijpers, B.: A clustering-based approach for discovering interesting places in trajectories. In: SAC, pp. 863–868 (2008)
9. Xiu-Li, Z., Wei-Xiang, X.: A clustering-based approach for discovering interesting places in a single trajectory. In: ICICTA, pp. 429–432 (2009)
10. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: WWW, pp. 791–800 (2009)