Discovering Personal Paths from Sparse GPS Traces

Changqing Zhou, Shashi Shekhar, Loren Terveen
Department of Computer Science and Engineering, University of Minnesota
200 Union ST SE, 4-192, Minneapolis, MN 55414
{czhou, shekhar, terveen}@cs.umn.edu

ABSTRACT

Personal paths capture "personal meaningful places" [13, 14] in temporal sequence. Knowledge of a user's paths enables novel and useful features for location-aware applications, e.g., traffic condition updates for commuting routes, carpool partner finding with similar commuting routes and schedules. Prior work has explored algorithms to discover "significant locations" [1,2] and "transportation routes" [6, 9] from GPS data, however, we know of no algorithms specifically designed for sparse GPS traces, which represent typical location datasets collected from GPS enabled mobile devices. In this paper, we report two spatio-temporal clustering algorithms, TDJ and R-TDJ, for discovering personal paths. Specifically, the algorithms are designed to address the noisy and sparse nature of GPS data. Our experiment results show that both TDJ and R-TDJ discovered meaningful spatio-temporal clusters to form personal paths. R-TDJ demonstrates better performance on sparse GPS data.

Keywords

Spatio-temporal data mining, ubiquitous computing, location-aware applications, clustering algorithms, paths discovery.

1. INTRODUCTION

A person's paths represent a person's daily routine. For example, a person's typical paths in a weekday may include the following stops: leaving home in the morning, dropping her child at school on the way to work, arriving work place, taking a break during lunch time, leaving work in the afternoon, picking up child from school on the way home, arriving home, taking child to sports activities after dinner, and going back home.

Personal paths capture "personal meaningful places" [13, 14] in temporal sequence. Represented in paths, the same physical place associated with different timestamps may imply different activities. For example, a family might go to a recreation center on Wednesday evening for a swimming class; this path might have associated actions like checking the traffic report, remembering to fix dinner early, bringing the swinging bag, etc. However, if the family goes to the same recreation center on Saturday morning for ice skating, different actions are relevant: bring skates and helmet, for example.

Knowledge of users' paths enables novel and useful features for location-aware applications. For example, people can get updates on traffic conditions for their commuting routes; a cognitively impaired person or his caretaker can be alerted when he departs from his usual routine; people can find carpool partners with similar commuting routes and schedules.

In this paper, we report two spatio-temporal clustering algorithms for discovering personal paths. Specifically, the algorithms are designed to address the noisy and sparse nature of GPS data. The rest of the paper is organized as follows. In the related work section, we report the related literature on "place" discovery approaches and algorithms. We then describe our algorithms: spatio-temporal clustering (TDJ) and relaxed spatio-temporal clustering (R-TDJ). In the experiment section, we present our experience with the algorithms on real GPS data collected from mobile devices. Finally, we briefly conclude our work.

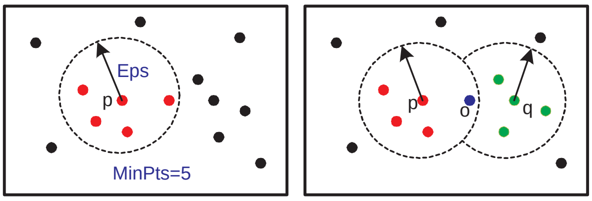
2. RELATED WORK AND OUR CONTRIBUTIONS

One of the early "place" discovery system is commotion [8], which consists of a device that constantly takes GPS readings. The loss of the signal is interpreted as a significant cue, namely that a building has been entered. However, this approach can not discover some meaningful places (such as a park or sidewalk cafe) because they may not cause any GPS signal loss.

The well-known K-Means clustering algorithm was used to learn a user's significant locations from location history data [1,2]. However, K-Means approach suffers the drawbacks such as required cluster number before clustering and sensitivity to noise data. In their study, a Markov model was used to predict the transitions between the places. Similarly, other machine learning models, such as Dynamic Bayesian Network [9] and hierarchical Dynamic Bayesian Network [6] were used to learn user's daily routines.

Many of the limitations of the K-Means clustering approach can be overcome with a density-based clustering approach. First, it can discover clusters of arbitrary shape. This is a significant improvement over K-Means, which favors symmetric shaped clusters (circles and spheres). Second, noise, outliers, or simply unusual points are less likely to participate in the final clustering results. Third, although density-based algorithms require density parameters (e.g., Eps and $MinPts$) as input, these parameters are less likely to need to change within a particular application. Finally, the density-based algorithms we describe always produce the same clustering given the same input. DBSCAN is a representative density-based algorithm [3, 10]. One issue with it is that it is very sensitive to the parameters Eps and $MinPts$. For some Eps and $MinPts$, the

This work was partially supported by grants from the NSF (IIS 03-07459 and IIS 03-08018). It also was partially supported by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under contract number DAAD19-01-2-0014. The content of this work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred.
WDM 2005 (1st International Workshop on Data Mining in conjunction with 8th Joint Conference on Information Sciences), July 21-26, 2005, Salt Lake City, Utah, USA.



(a) $Density - N(p)$ (b) $Density - Joinable$

Figure 1: Density-based join concept

algorithm will generate a large number of points within its density definition, each of which could be further used to generate its own density-reachable points. In such cases, it will use a lot of memory and slow down considerably. Our experience with these performance problems led us to develop a different density and join-based clustering algorithm: *DJ-Cluster* [13, 14].

Kang et al. [5] extracted places using a simple accumulative clustering algorithm from traces of location data collected using WI-FI technology. However, the high frequent data sampling (1 reading per second) could potentially degrade the performance and consume a significant amount of battery from mobile devices.

Mamoulis et al [7] presented algorithms to discover "periodic patterns", e.g., the same routes (approximately) over regular time intervals. In their approach, they divided sequence of locations into different set by the same interval T. For each set, they run a hash-based clustering method to obtain the initial clusters for future Aprio type of association rule mining. We argue that this equal temporal interval decomposition method may split real clusters into different sets. Besides, this approach is not suitable for sparse GPS data. The time a mobile user spent moving around typically account for a small portion of her daily routine. Equal temporal interval decomposition will generate a lot of clusters at the locations where the user is stationery. There are other related scalable algorithms in data mining literatures on pattern mining from very large historical spatiotemporal dataset [4, 11, 12].

This paper makes following contributions. First, it proposes density and join-based temporal clustering algorithm (TDJ) to discover personal paths. Second, it proposes a variation of TDJ with relaxed temporal constraints, R-TDJ, to address the noisy and sparse GPS data. Third, it formularizes the computational complexity of the algorithms. Last, it provides results of personal paths discovered from real GPS data collected from small mobile devices.

3. OUR APPROACHES

In this section, we describe two clustering algorithm to discover personal paths: join-based temporal clustering (TDJ) and TDJ with relaxed temporal constraints (R-TDJ). Both TDJ and R-TDJ are based on DJ-Clustering [13]: density and join-based. The algorithms treat time as the third dimension and take time elapse threshold of two points as the third input, $deltaT$, along with Eps and $minPts$. Fig. 1 shows the concept of density-based clustering [13].

3.1 The Algorithms: TDJ and R-TDJ

3.1.1 TDJ

Our density and join-based temporal clustering algorithm (TDJ) is based on DJ-Cluster [13]. The basic idea of TDJ is as follows. For each point, calculate its *neighborhood*: the neighborhood consists of points within distance Eps and time $deltaT$, under the condition that there are at least $MinPts$ of them. If no such neighborhood is found, the point is labeled noise; otherwise, the points are created as a new cluster if no neighbor is in an existing cluster, or joined with an existing cluster if any neighbhour is in an existing cluster. The algorithm is described below in Algorithm 1.

Algorithm 1 TDJ

```
while exist unprocessed point $p$ from sample $S$ do
    Compute the density-based neighborhood $N(p)$ wrt $Eps$, $MinPts$ and $deltaT$.
    if $N(p)$ is null then
        Label $p$ as noise.
    else if $N(p)$ is density-joinable to an existing cluster then
        Merge $N(p)$ and all the density-joinable clusters.
    else
        Create a new cluster $C$ based on $N(p)$.
    end if
end while
Return Path($C_0$, $C_1$, ..., $C_i$, ..., $C_n$), $C_i$ are ordered by time
```

3.1.2 R-TDJ

R-TDJ was designed to capture repetitive events, with short elapse times at the same location. For example, a person may drive through the same fast food restaurant a couple of times during a day. However, because each time this person only stays there for a short period of time, it does not accumulate enough location readings to meet the $minPts$ constraint to form a cluster. To handle this situation, we propose relaxed temporal constraint strategy. The algorithm is described below in Algorithm 2.

Our relaxed temporal constraint strategy, illustrated in step 4 in the algorithm, is not about increasing the value the $deltaT$, which will not help in this situation because the clustering is already bounded by the spatial constraint and a small increasing of $deltaT$ value will not include more points. At the same, we can not rely on decreasing $MinPts$ to form a cluster from these points because it will generate a lot of false positive clusters at the other places, such as traffic stops.

Relaxed temporal constraint strategy considers points that fall in the same spatial constraint but in a much larger temporal window. The window will be large enough to capture a different visit at the same location. Using the above example, suppose there is no enough points in the morning visit to form a cluster, our strategy will relax the temporal constraint to count the evening visit points against $MinPts$. That is, the points from morning visit and evening visit will be counted together against the $MinPts$ constraint, so that a morning cluster will be formed, but the morning cluster will not be merged with the evening cluster.

3.2 Eps, $MinPts$, $deltaT$ and $rDetltaT$

Comparing with DJ-Cluster, there is one more input parameter in TDJ and R-TDJ, $deltaT$, besides Eps and $MinPts$. These three parameters together determine the density of the location and temporal neighbors and thus the size and shape of the clusters. To discover smaller, "skinnier" and a larger number of clusters, one can decrease 3 parameters parameters.

The values of Eps, $MinPts$ and $deltaT$ to TDJ may be determined by specific applications. In an application to discover personal paths from GPS data, Eps may be set to approximate the uncertainty in GPS readings, e.g., to 20 meters. Suitable values for $MinPts$ range from 3 to 10; higher values mean that clusters must be more dense to form. Different values for $deltaT$ will not

Algorithm 2 R-TDJ

```
while exist unprocessed point $p$ from sample $S$ do
    Compute the density-based neighborhood $N(p)$ wrt $Eps$, $MinPts$ and $deltaT$.
    if $N(p)$ is null then
        Compute the density-based neighborhood with relaxed temporal constraint $RN(p)$ wrt $Eps$, $MinPts$, and $rDeltaT$.
        if $RN(p)$ is null then
            Label $p$ as noise.
        else
            Create a new cluster $C$ based on $RN(p)$.
        end if
    else if $N(p)$ is density-joinable to an existing cluster then
        Merge $N(p)$ and all the density-joinable clusters.
    else
        Create a new cluster $C$ based on $N(p)$.
    end if
end while
Return Path($C_0$, $C_1$, ..., $C_i$, ..., $C_n$), $C_i$ are ordered by time
```

significantly change the spatial compactness of clusters, however, if there are abundant of points in a cluster meeting the spatial constraints, decreasing $deltaT$ may split the clusters into smaller ones with different time periods. In a location-aware application running on GPS enabled cell phone, a typical $deltaT$ can be set as 5 - 10 minutes.

$rDetltaT$ is introduced for R-TDJ to relax the temporal constraint of a cluster. Greater $rDetltaT$ values will lead to more relaxed temporal constraint, thus find larger number of less repetitive events. A typical value could be set as 24 hours for daily repetitive events, or 168 hours for weekly ones.

3.3 Computational Complexity

Our current TDJ and R-TDJ are main-memory implementations. We can analyze it in two steps.

First, computing the neighborhood of a point is $O(n^2)$ without a spatial index, or $O(n \log n)$ with an R-tree index.

Second, another major cost is the join computation for each point's neighborhood with existing clusters. This is $O(n^2)$ without a spatial index, or $O(n \log n)$ with an R-tree index.

Thus overall, the complexity of both the algorithms is $O(n \log n)$ with an R-tree index.

4. EXPERIMENTAL EVALUATION

4.1 Location Data Collection

The first author of this paper carried a GPS-enabled mobile phone for three weeks as he went about his daily activities in the Minneapolis – Saint Paul metropolitan area in the United States. His normal transportation mode was driving a personal car. His routine included commuting to work, frequent visits to the University of Minnesota campus, and various errands.

He attempted to keep the phone with him and on at all times. The phone ran the Accutracking service, which was configured to take a GPS reading every minute. During the 20-day experiment period, 3,469 GPS readings were collected. On average, this was about 173 locations per day, or nearly three hours worth of location data. These location data are visualized on the map in Fig. 2.

4.2 Results

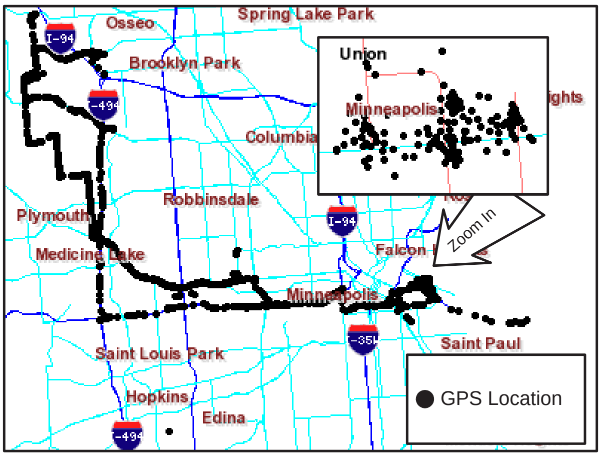


Figure 2: A personal location dataset: 20-day experientment period, most of the locations folow major roads.

We ran both TDJ and R-TDJ algorithms on one day worth of GPS traces from the collected personal location dataset.

4.2.1 TDJ

Input: $MinPts$ = 10, Eps = 10 and $deltaT$ = 5min.

Discovered path: *Leave home (8:21) → arrive work (8:59) →arrive school(11:42) → run errand (12:33)→ arrive work (12:52)→ leave work (17:29)→ stop daycare (17:49)→ stop home (18:13)→ arrive swimming pool*, shown in Fig. 3.

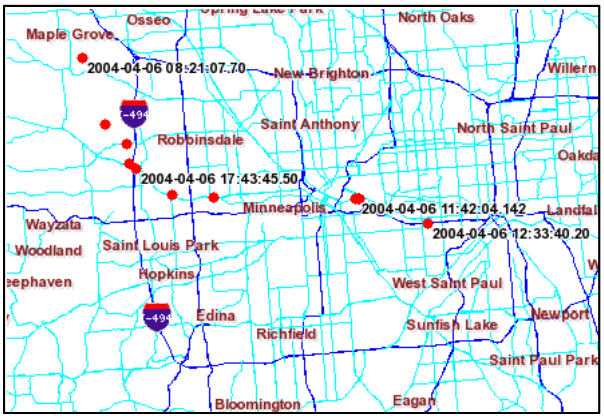


Figure 3: TDJ: the red dots represent discovered clusters/places of the path. The places form a path when ordered in time sequence.

4.2.2 R-TDJ

Input: $MinPts$ = 10, Eps = 10 and $deltaT$ = 5 min, $rDeltaT$ = 24 hours.

Discovered path: *Leave home (8:21) → stop daycare (8:43) → arrive work (8:59) → leave work (11:24) →arrive school(11:42) → run errand (12:33)→ arrive work (12:52)→ leave work (17:29)→ stop daycare (17:49)→ stop home (18:13)→ arrive swimming pool (18:29)→ leave swimming pool (19:43)→ arrive home (19:50)*, shown in Fig. 4. We also visualized the path in 3-D, shown in Fig. 5.

4.2.3 Discussion

Both TDJ and R-TDJ generate meaningful spatio-temporal clusters. However, R-TDJ returned a complete path on that day, while

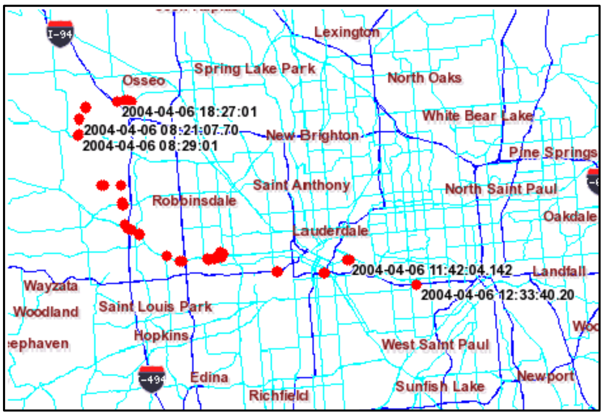


Figure 4: R-TDJ: the red dots represent discovered clusters/places of the path. The places form a path when ordered in time sequence.

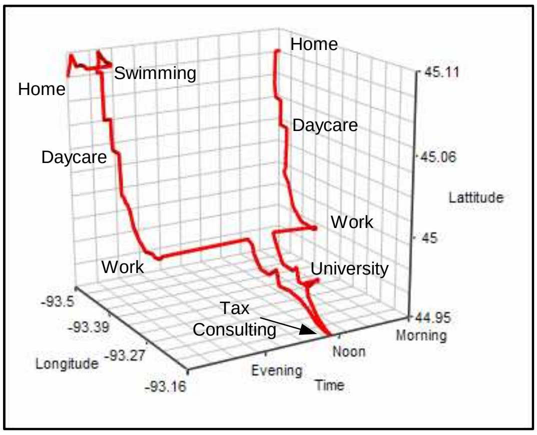


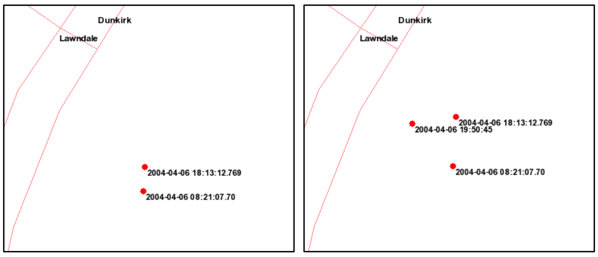
Figure 5: 3-D visualization of path generated from R-TDJ

TDJ missed some activities or places, e.g., morning stop at daycare, leaving work at lunch time, leaving swimming pool and arriving home for the night. These activities were discovered by R-TDJ because of the relaxed temporal constraint.

We use multiple instances of home discovered in the path as an example to explain the difference of the algorithms, shown in Fig. 6. During that day, the author visited home 3 times, which resulted in 3 "chunks" of GPS readings. GPS readings in each "chunk" are spatially and temporally close to each other. 2 clusters were formed in the morning and evening, respectiely, e.g., *leave home (8:21)*, and *stop home (18:13)*, but there were not enough of points to meet the $MinPts$ constraint to form a cluster at night, e.g., *arrive home (19:50)*. In this case, R-TDJ relaxed the time constraint to allow GPS readings from other visits at home (still temporally constrained by $rDeltaT$ window) to be counted against the $MinPts$. R-TDJ however keeps *arrive home (19:50)* as a different cluster from the others: *leave home (8:21)* and *stop home (18:13)*.

5. CONCLUSION

In this paper, we developed two spatio-temporal clustering algorithms, TDJ and R-TDJ, to discover personal paths. Specifically, the algorithms were designed to address the noisy and sparse nature of GPS data. Our experiment results showed that both TDJ and R-TDJ discovered meaningful spatio-temporal clusters to form personal paths. R-TDJ demonstrated superior performance on sparse GPS data.



(a) TDJ: leave home at 8:21, stop home at 18:13

(b) R-TDJ:leave home at 8:21, stop home at 18:13, and arrive home at 19:50

Figure 6: Instances of home discovered by TDJ and R-TDJ.

6. REFERENCES

[1] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with GPS. In *Proc. IEEE 6th Intl. Symp. on Wearable Comp.*, 2002.

[2] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multipleusers. *Personal and Ubiquitous Computing*, 7:275–286, 2003.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. KDD*, 1996.

[4] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In *Proc. ICDE*, 1999.

[5] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. In *Proc. WMASH*, 2004.

[6] L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proc. AAAI*, 2004.

[7] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proc. KDD*, 2004.

[8] N. Marmasse and C. Schmandt. Location-aware information delivery with commotion. In *Proc. HUC*, 2000.

[9] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *Proc. UbiComp*, 2003.

[10] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2:169–194, 1998.

[11] I. Tsoukatos and D. Gunopulos. Efficient mining of spatiotemporal patterns. In *Proc. SSTD*, 2001.

[12] M. Vlachos, D. Gunopulos, and G. Kollios. Discovering similar multidimensional trajectories. In *Proc. ICDE*, 2002.

[13] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen. Discovering personal gazetteers: An interactive clustering approach. In *Proc. ACMGIS*, 2004.

[14] C. Zhou, P. Ludford, D. Frankowski, and L. Terveen. An experiment in discovering personally meaningful places from location data. In *Proc. CHI, Extended Abstract*, 2005.