Mining Individual Life Pattern Based on Location History

Yang Ye[#*1], Yu Zheng[*2], Yukun Chen[*3], Jianhua Feng[#4], Xing Xie[*5]

[#]*Dept. Of Computer Science and Technology, Tsinghua University*
Beijing, 100084, P.R. China
[1]yey05@mails.tsinghua.edu.cn
[4]fengjh@tsinghua.edu.cn

[*] *Microsoft Research Asia*
4F, Sigma Building, No.49 Zhichun Road, Haidian District, Beijing 100190, P. R. China
[2.3.5] {yuzheng,v-yukche, xing.xie}@microsoft.com

Abstract— **The increasing pervasiveness of location-acquisition technologies (GPS, GSM networks, etc.) enables people to conveniently log their location history into spatial-temporal data, thus giving rise to the necessity as well as opportunity to discovery valuable knowledge from this type of data. In this paper, we propose the novel notion of individual life pattern, which captures individual's general life style and regularity. Concretely, we propose the life pattern normal form (the LP-normal form) to formally describe which kind of life regularity can be discovered from location history; then we propose the LP-Mine framework to effectively retrieve life patterns from raw individual GPS data. Our definition of life pattern focuses on significant places of individual life and considers diverse properties to combine the significant places. LP-Mine is comprised of two phases: the modelling phase and the mining phase. The modelling phase pre-processes GPS data into an available format as the input of the mining phase. The mining phase applies separate strategies to discover different types of pattern. Finally, we conduct extensive experiments using GPS data collected by volunteers in the real world to verify the effectiveness of the framework.**

I. INTRODUCTION

Nowadays, the development in location-acquisition technologies and its embedding into people's daily life results in a novel type of spatial-temporal data, which traces individual location history and can be collected by the wireless network infrastructures. For instance, when mobiles phones are connected to GSM network, they left positioning logs together with the timestamp of each log point. Likewise, GPS-embedded portable devices can also record the latitude-longitude position at every moment when exposed to a GPS satellite. The increasing availability of individual location history data bring us challenges as well as opportunities to discover valuable knowledge from the raw data.

On this topic, some literatures aim at performing traditional data mining tasks on spatial-temporal data, like classification [1], clustering [2], pattern mining [3], [4] and outlier detection [5]. In the meantime, some techniques have been proposed to discovery higher level knowledge from individual GPS data [15], including detecting significant locations of an individual, predicting the movement destination [6], [11], recognize individual mobility [14], etc.

However, the first class of research treats location history data as general spatial-temporal trajectory; thereby loses some of their inner properties. For instance, each log point in the location history contains absolute time spot. However, either Temporally Annotated Sequences in [7] or Trajectory Pattern Mining in [3] discards this absolute time information and just calculate the time interval between two points as time annotation. While current works in the second class typically mine first-level knowledge about position and mobility from location history, like significant places, possible destinations, attribute of mobility like stationary and walking.

Since location history data are individually generated, given the close relationship between people's daily life and geographic locations, we claim that one's general life style and regularity can be discovered from his/her location history. Resembling traditional definitions of frequent pattern in transaction database [8], we term individual's general life style and regularity *life pattern*. In contrast to the first level knowledge about position and mobility, life patterns represent a higher level knowledge drawn from location history data.

The discovery of life pattern has a manifold of application scenarios. To illustrate, life pattern reflects the regularity of one individual, thus can help people better learn their way of life; it can also be embedded into location recommender system, context-based computing, precise advertising, computer-aided schedule and route arrangement. For instance, if Tom's life pattern about the time he goes to work is discovered, his intelligent cell phone can automatically help arrange the travelling route according to that day's traffic condition. If Tom's general body-building time and place is discovered, intelligent advertising system may choose that moment to cast health products advertisement to his mobile equipment. Also, computer-aided schedule and route arrangement system can intelligently advice Tom to arrange the time, route or position of new activities, given his general life pattern. In the meantime, through collecting and analysing life patterns of multiple individuals, a lot of statistical and mining work can be done to discovery valuable knowledge about social trends and generalities.

This paper is motivated by the increasing availability of individual GPS data and the usefulness of life pattern. We aim

at mining individual life pattern from individual generated GPS data. Concretely, the contribution of this paper lies in following aspects:

- We propose the novel notion of individual life pattern. We introduce the life pattern normal form (*LP-normal form*) to formalize the expression of life patterns and facilitate the mining framework. Life pattern emphasize *significant places* in one's daily life, because these places reflects his/her typical life activities. An *atomic life pattern* corresponds to one significant place. LP-normal form takes into consideration several properties to combine atomic patterns into complicated ones. We also introduce life associate rules as a special type of pattern.
- We propose the LP-Mine framework to effectively extract life patterns from raw GPS data. In the first phase, through two steps of modelling: *stay point detection* and *stay point clustering*, LP-Mine transformed individual GPS data into the *location history sequence* as the input of the mining phase. In the second phase, LP-Mine applies separate methodologies for different types of patterns, including temporal sampling and partition, frequent itemset and sequence mining, corset discovery, etc.
- We conduct extensive experiments using GPS data collected by volunteers over a period of 6 months to verify the effectiveness of LP-Mine framework. We inject LP-Mine into "*GeoLife*", a GPS-log-driven application on Web Map, to visualize the mined patterns, thus we can conduct user study to verify the interestingness and representativeness of mined patterns.

The rest of the paper is organized as follows. In Section II, we survey related work and point out the differences between ours and others. In Section III, we detailedly analyse several aspect with regard to life patterns; then propose the life pattern normal form to formalize the definition of life patterns. In Section IV and V, we respectively detail the "modelling" and "mining" phase of LP-Mine. The experimental evaluations and discussions are provided In Section VI. Our conclusion and outlook on the future work are given in Section VII.

II. Related Work

A. Traditional Frequent Pattern Mining and Association Discovery

As a major task of data mining, frequent pattern mining and associate rule discovery have been exhaustively studied in the last decade. In [8], the idea of frequent pattern and associate rules is first introduced based on *basket analysis*. A basket corresponds to a transaction containing different items. Given a threshold *s* and a transaction dataset *D*, the mining task is to discovery itemsets whose support (percentage of transactions in *D* containing the itemset) is greater than *s* as frequent patterns. *Associate rule* is extension of frequent pattern. An associate rule "$A \rightarrow B$" carries the semantic that if people buy items in A, they tend to buy items in B. Associate rule discovery uses frequent itemset mining as the first step. Frequent sequential pattern is first addressed in [9]. Items forming a sequential pattern must obey certain order and the mining task is to discovery frequent subsequence from a sequence dataset.

Early frequent pattern mining algorithms tend to generate exponential number of patterns, most of which are valueless. Therefore, *closed* itemset and subsequence are proposed. A pattern is called "closed" if none of its superset or super-sequence is frequent. Closed patterns retain valuable information and are of polynomial number; therefore they are widely applied. Representative algorithms for closed itemset and sequence mining are closet+ [10] and CloSpan [11].

Frequent pattern mining is extended to various types of data and applied in different contexts. J. Han. [19] presents a comprehensive survey of frequent pattern mining categories, techniques and applications. Traditional frequent pattern mining cannot be directly applied to trajectory data because of the fussiness of space (no two point in trajectory data is exactly the same) and the introduction of temporal information.

B. Trajectory Data Mining

On applying traditional data mining tasks to spatial-temporal data or trajectory data, several paradigms have been proposed to solve the fuzziness in locations and utilize the temporal information in original data. On mining sequential pattern on spatial data, [4] defines pattern elements as spatial regions around frequent line segments. First, original sequence is transformed into a list of sequence segments; then frequent regions are detected in a heuristic way; finally patterns are detected using a substring tree structure. On extending sequence pattern mining, the temporal annotation sequence (*TAS*) is first introduced in [7] to represent the transmission time between sequence elements. However, [4] does not assume temporal information and [7] does not assume the fussiness of sequence elements. In [3], both space (i.e., the regions of space visited during movements) and time (i.e., the duration of movements) are taken into consideration to define a trajectory pattern.

In [2], [1], [5], the same definition of perpendicular, parallel and angular distance between two trajectory partitions (t-partitions) is proposed. Then a similar partition phase is employed to divide original trajectory into set of t-partitions based on the notion of minimal descriptive length (MDL). Finally, different mining tasks: clustering, classification and outlier detection are performed on the resulted t-partition set. All these trajectory data mining works deals with the raw data.

Most existing techniques on trajectory data mining can be applied to general trajectory data like *hurricane track* and *animal movement*. Even when applied to individual GPS data, they typically cannot discovery knowledge about personal life. In contrast, our work focuses on individual GPS data and discovers individual life regularity as life patterns.

C. Mining Location History

There are also several works on mining location history based on GPS data. On mining individual location history, [13] focuses on detecting significant locations of a user, predicting user's movement among these locations. [14] deals with recognizing user-specific activities among significant locations. [15] introduces a hierarchical Markov model that can learn and infer a user's daily movements through an urban community. [16] presents a method of learning a Bayesian model of a traveler moving through an urban environment. This method simultaneously learns a unified model of thetraveler's current mode of transportation as well as his most likely route in an unsupervised manner. [6] uses a history of a drivers destinations, along with data about driving behaviors, to predict where a driver is going as a trip progresses

There are also works on mining multiple users' location history. [17] develops a system called LOCADIO, which uses Wi-Fi signal strengths from existing access points measured on the client to infer user mobility such as, stationary and walking, etc.. [18] conducts similar work, while based on GSM network. Zheng et al. [15] aim to infer users' transportation mode like walking and driving, etc., based on GPS trajectories of 60 individuals. In [19], a hierarchical-graph-based similarity measurement is developed to mine similarity between different users from location history.

These techniques and systems generally discover knowledge about mobility, like positioning, destination, way of moving, etc. In contrast, we aims at mining individual life regularity.

III. Problem Formalization

A. Preliminary Analysis

To formalize the definitions of life patterns, we first deliberate several aspects of life patterns in this section.

1. **(*Temporal Granularity and Condition*)** The atomic temporal observation unit of life is "day". This should be attributed to the daily-repetitious natural of human activity. While different scale of temporal unit (*temporal granularity*) correspond to patterns of different semantics. For instance, using "day" as the unit, life patterns like
   "*Tom visits the cinema once a week*" (1)
   can never be discovered. In the meantime, there are also life patterns and associations with specific *temporal conditions*, such as "*on Mondays*", "*on work days*". These conditions are intuitive because the individual may have different life style in different type of days.

2. **(*Significant Places*)** Life pattern emphasizes the *significant places* in one's GPS record while ignores the transition between these places (in fact the transition can also be viewed as one type of life pattern, which we do not include in the framework and leave as future work). Because the significant places, like schools, companies and hospitals, can represent the individual's typical activity. One (simplest) life pattern could be:
   "*In 70% of the days, Tom visits Sigma Building*" (2)

3. **(*Sequentiality*)** We shall take into consideration not only the isolated significant places, but also the order in which the individual visits them. Life patterns without and with sequentiality correspond to their counterparts in traditional pattern mining [8], [9]. An typical sequential life pattern could be:
   "*In 50% of the days, Tom takes this route:* $Tsinghua\ Univ. \rightarrow Sigma\ Bld. \rightarrow Wal-Mart$" (3)

4. **(*Timestamp-Annotation and Timespan-Annotation*)** We may care about not only the places themselves, but also the individual's arrival time, departure time (*timestamps*), and duration time (*timespan*) of the significant places, because these timestamps and timespan give valuable description about the individual's life style. An example of this type of life patterns is:
   "*In 63% of the days, Tom arrives at Sigma Bld. between 8:50 a.m. to 9:10 a.m. and stays for more than 3 hours and less than 5 hours*" (4)

5. **(*Conditional Life Pattern*)** We may also concern about some pattern-conditioned life pattern, which use another life pattern to constrain the observation unites, like:
   "*Among the days on which Tom visits Sigma Bld., in 90% of them he arrives between 8:50 a .m. to 9:10 a.m. and stays for more than 3 hours and less than 5 hours"* (5)
   The condition and consequence may also refer to different significant places. In fact, they can be arbitrary non-conditional life pattern, like:
   "*Among the days on which Tom takes this route:* $Tsinghua\ University \rightarrow Sigma\ Building$, *in 60% of them he also arrives at Wal-Mart between 5:00 p.m. to 5:30 p.m.*" (6)

6. **(*Life Associate Rule*)** Life pattern (5) and (6) should be treated separately. The condition and consequence of (5) share the same location (Sigma Building) while those of (6) do not. (5) indicates the regular arriving time while (6) suggests some association between "taking the route: $Tsinghua\ University \rightarrow Sigma\ Bld.$" and "arriving at Wal-Mart between 5:00 p.m. to 5:20 p.m.".

   In fact, Patterns (1) through (6) are real cases drew from our experiments and "Tom" is an anonym of an Tsinghua University student who has an internship at MSRA located at Sigma Building. Pattern (1) through (5) seem trivial while (6) is somewhat interesting. Indeed, the Walmart supermarket is near Sigma Building and Tom tends to go shopping, or have dinner around Wal-Mart after work.

   We term the first type of conditional life pattern "*temporal-knowledge life pattern*", which concerns about temporal knowledge of given places. While the second type of pattern give rise to a "*life associate rule*", which we treat as another type of pattern. It suggests implied association between different life regularities. The definition of life associate rule is not simple extension from traditional associate rule in that the condition and

consequence can take arbitrary non-conditional life pattern, like in pattern (6).

B. The LP-Normal Form

Summing up previous analysis, we formally present the definition of life patterns in this section. Figure 1 gives the architecture of life patterns.

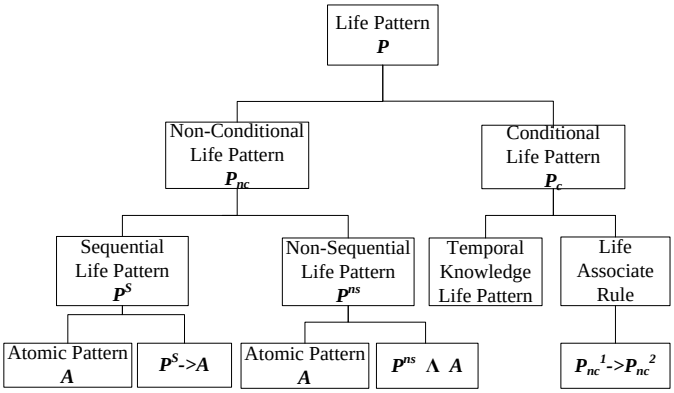


Fig. 1 Hierarchy of life patterns

A life pattern P can be either non-conditional (P_{nc}) or conditional (P_c). A conditional life pattern can be interpreted as one non-conditional pattern given another non-conditioned pattern. That is

$$P := P_c \parallel P_{nc} \quad (1)$$

$$P_c := P_{nc}^1 \mid P_{nc}^2 \quad (2)$$

A non-conditional life pattern can be either a non-sequential pattern (P^{ns}) or a sequential pattern (P^s).

$$P_{nc} := P^s \parallel P^{ns} \quad (3)$$

To define non-sequential and sequential life patterns, we introduce "*atomic life pattern*", which refers to visiting single significant place, with or without timestamp/span annotations. An atomic life pattern A is of the form:

$$A := visit\ (X).(?arv([t_1, t_2])).(?stay([\tau_1, \tau_2]) \quad (4)$$

Here the symbol ? means A can either be timestamp/span annotated or not. t_1, t_2 are two timestamps and τ_1, τ_2 are lengths of two timespans. Note we do not need a "leave time" annotation because it can be decided by the arrival timestamp and stay timespan.

There are two operators which combine atomic life patterns into a more complex one: the "*and*" operator "∧" and the "*sequence*" operator " → ". The former generates non-sequential life pattern P^{ns} and the latter generates sequential life pattern P^s

$$P^{ns} := A \parallel P^{ns} \wedge A \quad (5)$$

$$P^s := A \parallel P^s \rightarrow A \quad (6)$$

We don't assume non-conditioned life patterns of the form $P^s \wedge P^{ns}$. Because their semantic meaning, one's lifestyle that s/he both visits some places without order constraint and visits some other places follow a certain order, is unclear and not useful in applications.

Each pattern P is associated with a *support* value s, which denotes the percentage of temporal observation units when P is satisfied. Thus, a life pattern can be represented as

$$(P, s) \quad (7)$$

A conditional life pattern $P_c = P_{nc}^1 \mid P_{nc}^2$ naturally gives rise to a *life associate rule* $R: P_{nc}^2 \rightarrow P_{nc}^1$. Like in traditional associate rules, the *confidence* of R, denoted $c(R)$, is defined as the conditional probability of P_{nc}^1 given P_{nc}^2, thus equals the support of P_c. The *support* of R, denoted $s(R)$, equals the probability that P_{nc}^1 and P_{nc}^2 both happen. We have:

$$c(R) = \Pr[P_{nc}^1 \mid P_{nc}^2] = \frac{\Pr[P_{nc}^1,\ P_{nc}^2]}{\Pr[P_{nc}^2]} = \frac{s(R)}{s(P_{nc}^2)} \quad (8)$$

In sum, a life associate rule can be represented as:

$$(P_{nc}^2 \rightarrow P_{nc}^1,\ s,\ c) \quad (9)$$

Having formalized the definition of life patterns, the mining task can be described as: given individual GPS data and a support threshold s and confidence threshold c, discover individual life patterns with support (percentage of temporal observation unites containing the pattern) greater than s. For life associate rules, the confident should also be greater than c.

In following two sections, we shall concretely describe the LP-Mine framework towards mining life patterns presented in this section. Section IV details the modeling phase, which pre-processes raw GPS data into an available format, as the input of the mining phase described in Section VII.

IV. Data Pre-processing

A. Preliminary---GPS Log and GPS Sequence

The data collected by the GPS devices are of the GPS log form, which is a sequence of GPS points $P = \{p_1, p_2, p_3, ..., p_n\}$. Each point $p_i \in P$ contains the longitude ($p_i.Lngt$), latitude ($p_i.Lat$) and timestamp ($p_i.T$). As depicted in the left part of Figure 2. We can connect GPS points according to their time series into a GPS trajectory.

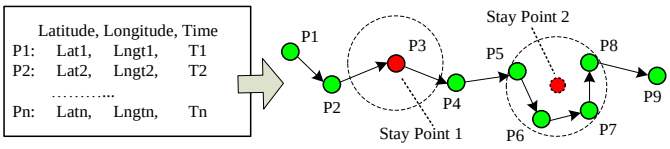


Fig. 2 GPS log and stay points

Since GPS point contains no semantic meaning like the spot name or attribute of places, we first need to extract significant places based on the spatial and temporal values of GPS points.

B. First Level Modelling---From GPS Sequence to Stay Point Sequence

We introduce the notion of stay points. A stay point S represents a geographic region in which the user stays for a

while. Therefore, each stay point carries its semantic meaning. For instance, the living and working places, the restaurant and shopping mall we visit, the spot we travel, etc.

We clarify two types of stay points as depicted in the right part of Figure 2. In the first case like *stay point 1*, the individual maintains stationary at *P3* for over a time threshold. This type of stay points generally occur when the individual enters a building and loses the satellite signal until returning to the outdoors. In the second case like *stay point 2*, the individual wanders around within a spatial region for over a time threshold. We use the mean longitude and latitude of the GPS points within the region to construct a stay point. Generally, this type of stay points occur when the individual wanders around some places, like a park, a campus, etc.

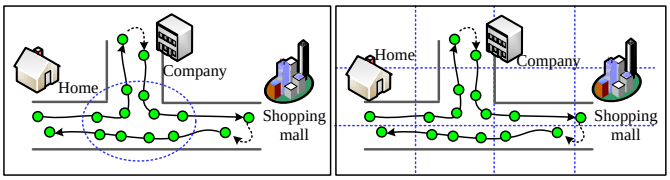
Figure 3 depicts the algorithm we apply to extract stay points from GPS data. We iteratively seek the spatial region in which the individual stays for a period over a threshold. For instance, in the experiments, a stay point is detected if the individual spends more than 30 minutes within a range of 200 meters. Note that each extracted stay point retains the temporal information: the arrival time ($S.arvT$) and the leaving time ($S.levT$) respectively equals the timestamp of the first and last GPS point constructing this stay point.

Algorithm StayPoint_Detection(*P, distThreh, timeThreh*)

Input: A GPS log *P*, a distance threshold *distThreh*
and time span threshold *timeThreh*
Output: A set of stay points *SP*={*S*}
1. *i*=0, *pointNum* = |P|; //the number of GPS points in a GPS logs
2. **while** *i* < *pointNum* **do**,
3. *j*:=*i*+1;
4. **while** *j* < *pointNum* ***do***,
5. *dist*=Distance(p_i, p_j); //calculate the distance between two points
6. **if** *dist* > *distThreh* **then**
7. $\Delta T=p_j.T-p_i.T$; //calculate the time span between two points
8. **if** ΔT>*timeThreh* **then**
9. *S.coord*=ComputMeanCoord({p_k | *i*<=*k*<=*j*})
10 *S.arvT*= $p_i.T$; *S.levT*=$p_j.T$;
11. *SP*.insert(*S*);
12. *i*:=*j*; break;
13. *j*:=*j*+1;
14. **return** *SP*.

Fig. 3 Stay points detection Algorithm

The reason we detect stay points in this way instead of directly clustering the GPS points, or using grid-based partition method to extract ROI (regions of interest) [3] lies in several aspects. For clustering GPS points as demonstrated in Figure 4-A, the first type of stay points will be lost, because the devices lose satellite signals indoors like homes and shopping malls. These places have few GPS points, therefore cannot satisfy the density condition to be detected by clustering. In the meantime, some regions like road crosses that a user iteratively passes will have a lot of GPS points. Although not containing valuable semantic information, they may be detected by clustering techniques. For grid based techniques as shown in Figure 4-B, the boundary problem will also results in missing significant places, because GPS points corresponding to the same place falls in different grids.



A) Clustering-based detection **B) Grid-based detection**

Fig. 4 Other possible stay points detection algorithms

C. Second Level Modelling ---From Stay Point Sequence to Location History Sequence

When stay points are detected, we use the stay point sequence $S = \{s_1, s_2, s_3, \dots, s_n\}$ to represent the individual's location history. Each stay point s_i corresponds to some significant place and $s_i.arvT$ and $s_i.levT$ correspond to the time of arriving and leaving this place.

However, this sequence still cannot be directly applied to mining life pattern. Actually no two stay points have the same spatial coordinates because of the fussiness of locations. For instance, different days' stay points for the place "company" are not identical, although they are very close to each other. Thus we need a second level modelling to group up different stay points with the same semantic meaning.

To address this, we apply density-based clustering as demonstrated in Figure 5. All individual's stay points are put into a dataset and clustered into several geographical regions. In comparison to partition-based clustering methods like *k*-means and grid-based methods like STING, density-based methods are capable of detecting clusters with irregular structure. For instance, in the experiments, we adopt the OPTICS clustering method which has two parameters, number of points (*NoP*) and distance threshold (*disThre*), when there are at least *NoP* points within *disThre* of a already clustered point, the new points are added to the cluster. In this way, a cluster is formed as a closure of points. OPTICS suits well in our application. Stay points of the same place are directly clustered into a density-based closure. In the meantime, clusters with valuable semantics may also be detected, such as a set of restaurants or travelling spots near a lake.

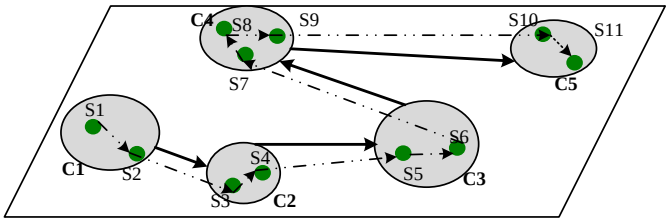


Fig. 5 Clustering stay points

After clustering the stay points, we transform the individual stay point sequence into the location history sequence $C = \{c_1, c_2, c_3, \dots, c_n\}$. Each stay point is substitute by the cluster it pertains to. In the meantime, the arriving time and leaving time of this stay point are retained and associated with the cluster. In this way, we will have location history records for different days' visiting of the same place like company or restaurant. The temporal value will be use in mining timespan-annotated and timestamp-annotated life patterns.

V. MINING INDIVIDUAL LIFE PATTERN

Through the abstraction phase, LP-Mine transforms GPS log data into individual location history sequence, which is a sequence of stay point clusters with timestamp annotation.

Figure 6 presents an overview of LP-Mine framework. Through data pre-processing, GPS log files are transformed into an individual location history sequence. We apply separate methodologies to mine different type of patterns.

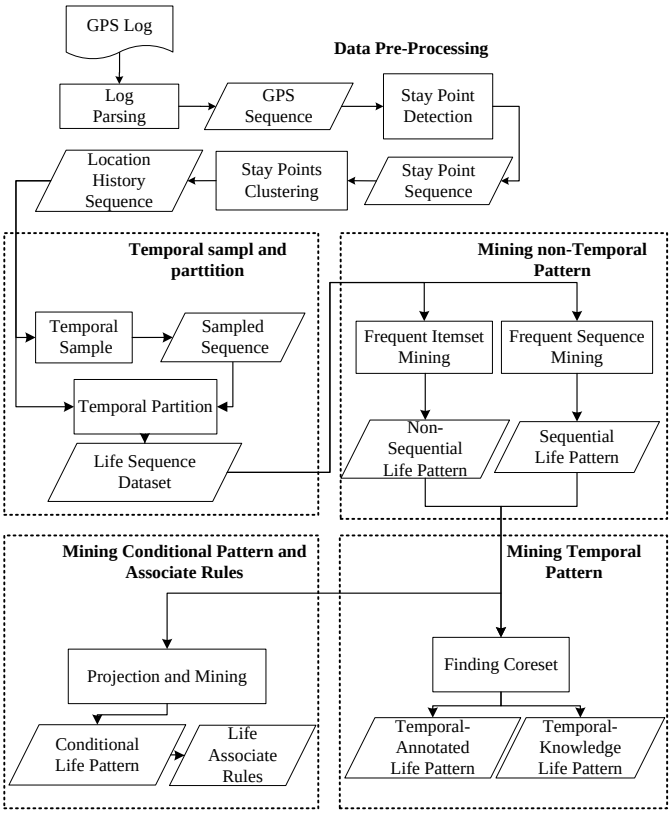


Fig. 6 Architecture of LP-Mine

A. Temporal Sampling and Temporal Partition

Temporal sampling deals with temporal condition discussed in Section III-A. For life patterns with specific temporal condition like "on Mondays", "on Weekdays", etc., original location history sequence is sampled according to the condition. For patterns and associations without temporal condition, this step is omitted.

Temporal partition corresponds to temporal granularity. Original location history sequence is partitioned into sub-sequences according to the specific granularity like "day", "week", etc. In this way, we construct a *life sequence dataset* $D^s = \{d_1, d_2, d_3, \dots, d_n\}$. Each d_i in D^s corresponds to the life sequence of one day, or one week or etc., according to the granularity. Mining work is then performed on D^s.

B. Mining Non-Sequential and Sequential Life Patterns

For non-sequential life patterns, we apply the closet+ [10] frequent pattern mining algorithm. For each d_i in D^s, the sequential property is ignored and each d_i is treated as a set of significant places. Closet+ applies several strategies including the hybrid tree projection method, the item skipping technique, etc. to effectively retrieve frequent closed itemset.

For sequential life patterns, we apply the CloSpan[11] algorithm. By mining frequent closed subsequences only, CloSpan produces a significantly less number of discovered sequences than the traditional full set methods while preserving the same expressive power.

C. Mining Temporal-Annotated and Temporal-Knowledge Life Pattern

Previously mined life patterns are out of timestamp/span annotation. As discussed in Section III-A, based on one (sequential or non-sequential) non-temporal life pattern, two types of temporal pattern can be mined: *temporal-annotated* life pattern and *temporal-knowledge* life pattern. The former uses whole D^s as the mining base, such as pattern (4) in Section III-A; the latter considers temporal knowledge given fixed locations, like pattern (5).

Assume the base non-temporal pattern to be $P = A_1 \wedge A_2 \wedge \dots \wedge A_d$ or $P = A_1 \rightarrow A_2 \rightarrow \cdots \rightarrow A_d$ and there are totally m sequences in D^s containing P, denoted $d_{p1}, d_{p2}, \dots, d_{pm}$. The temporal annotations of all d_{pi}*'s* form a set of m points in $2d$-dimension space. (d dimensions are A_i*.arv* for each $1 \le i \le d$ and d dimensions are corresponding A_i*.stay*) Indeed, temporal annotation can be added on any subset of the *2d*-diemensions, but annotation on all *2d* dimensions, on all *d* timestamp dimensions or on all *d* timespan dimensions should be most meaningful and widely adopted.

Denote the temporal-annotated pattern from P to be P^{t1} and the temporal-knowledge pattern from P to be P^{t2}. Assume the relative support threshold for temporal-annotated life pattern to be s_1, since $s(P^{t1}) \le s(P)$; we should first mine non-temporal-annotated patterns with support at least s_1, for each P of them and corresponding m sequences containing P, each sequence with $2d$ temporal annotations can be viewed as a point in $2d$-dimension space. Thus the task can be described as:

"*Finding a hyper-rectangle in 2d (d, or other selected temporal-dimensions) – dimension space that contains at least* $s_1/s(P)$ *proportion of the m points while minimize the volume of the hyper-rectangle.*"

Here we define volume to be the sum of all edge's lengths. Unfortunately, this combinatory optimization problem is NP-complete. In our experiment, when the temporal granularity is "month", there are at most 18 sequences corresponding to one year and a half. m should be even smaller. We can just enumerate all the $\left(\left\lfloor \frac{s_1}{s(P)} \cdot m \right\rfloor^{m}\right)$ candidate solutions.

When the temporal granularity is "day" or "week", enumeration does not work. We resort to *corsets* [21] and geometric approximation via corsets [20].

A corset is a subset Q of point set U that approximates the original set. In our application scenario, consider the optimal hyper-rectangle with minimized volume. V_{opt}, and points $U' = U \cap V_{opt}$, here U is the set of m points in $2d$-space. According to [21], we can find a corset $Q \subseteq U'$ such

that (1) $|Q| = O(1/\varepsilon)$ and (2) the smallest enclosing hyper-rectangle of Q, if ε-expanded (all lengths multiplied by $(1 + 1/\varepsilon)$), contains at least $\left\lceil \frac{s_1}{s(P)} \cdot m \right\rceil$ points of U. Thus, we can enumerate all possible subsets of size $O(1/\varepsilon)$ as "candidates" for Q, and for each subset, compute its smallest enclosing hyper-rectangle, ε-expand the rectangle and check how many points of U it contains. The smallest candidate hyper-rectangle that contains at least $\left\lceil \frac{s_1}{s(P)} \cdot m \right\rceil$ points of U is the required approximation. The running time of this algorithm is $dn^{O(1/\varepsilon)}$.

Likewise, for temporal-knowledge life pattern and support threshold s_2, the mining base is all m sequences containing P, and we just need apply the same methodology to find the smallest hyper-rectangle containing at least s_2 proportion of all points.

D. Mining Conditional Life Pattern and Life Associate Rules

A conditional life pattern of the form $P_c = P_{nc}^1 \mid P_{nc}^2$ carries semantic meaning that a sequence contains P_{nc}^1 given that it contains P_{nc}^2. For mining conditional life patterns, we apply a project-and-mining procedure. Given a mined non-conditional pattern P_{nc}^2 and the set of sequences containing P_{nc}^2, denoted U. We construct the projected sequences set of U on P_{nc}^2, denoted U'. Namely, for each sequence in U, delete the elements corresponding to P_{nc}^2. Then another step of mining procedure is performed on U'.

Generating life associate rules from conditional life pattern is intuitive but more complicated in that not only support threshold s but also confidence threshold c should be considered. According to Equation (8) in Section III, for associate rule $R: P_{nc}^2 \rightarrow P_{nc}^1$, we have:

$$s(R) = c(R) \cdot \mathrm{s}(P_{nc}^2) \tag{10}$$

Thus

$$s(R) \leq \mathrm{s}(P_{nc}^2) \tag{11}$$

Therefore, in order to mine life associate rules with support no less than threshold s, we should first mine non-conditional patterns P_{nc}^2 with support no less than s. Secondly, find all conditional life patterns P_c with P_{nc}^2 as the condition and have support greater than c (Note the support of conditional life pattern equals the confidence of corresponding life associate rule). Finally, check if $s(P_c) \cdot \mathrm{s}(P_{nc}^2)$ is greater than s. If so, this life associate rule is retained.

E. Summary

Summing up previous discussions, we summarize LP-Mine's procedure of mining individual life pattern in this section.

- ***First level modelling.*** Through stay point detection, LP-Mine extracts the significant places from GPS sequence, and transforms individual GPS sequence into stay point sequence.
- ***Second level modelling.*** Through density-based clustering, stay points pertaining to the same or similar significant places are clustered up. LP-Mine transforms individual stay point sequence into location history sequence.
- ***Temporal sampling and partition.*** For given temporal condition and temporal granularity, individual life sequence dataset is constructed by temporal sampling and partition.
- ***Mining non-temporal life patterns.*** LP-Mine applies closet+ and CloSpan techniques respectively for mining (non-temporal) non-sequential life patterns and sequential life patterns from the life sequence dataset.
- ***Mining temporal-life patterns.*** Based on one non-temporal life pattern, LP-Mine mines temporal-annotated and temporal-knowledge life patterns using the same technique of geometric approximation through corsets.
- ***Mining conditional life patterns.*** Based on one non-conditional life pattern P_{nc}, LP-Mine mines conditional life pattern using a projection-and-mining step. A projected life sequence dataset U' is constructed deleting elements corresponding to P_{nc} from sequences containing the pattern. Then another step of pattern mining is performed on U'.
- ***Mining life associate rules.*** The procedure resembles that of mining conditional life pattern. While LP-Mine considers not only support but also confidence threshold. For $R: P_{nc}^2 \rightarrow P_{nc}^1$, $\mathrm{s}(P_{nc}^2)$ should be no less than threshold s. $s(P_{nc}^1 \mid P_{nc}^2)$ should be above threshold c. Finally, it's also checked $s(P_{nc}^1 \mid P_{nc}^2) \cdot \mathrm{s}(P_{nc}^2)$ is greater than s.

VI. EXPERIMENTS

In this section, we experimentally evaluate the effectiveness of LP-Mine framework. Firstly, we present the experimental setting including the GPS devices we used, the volunteers we summoned, the data collected and the assignment of some parameters in the framework. Then, we conduct both objective and subjective experiments. In the objective experiment, we equally divide individual GPS data into two parts and verify how the second part of data matches patterns mined from the first part. This experiment aims at measuring the predictability of LP-Mine. In the subjective experiment, we visualize the mine patterns into GeoLife, a GPS-log-driven application on Web Map, and perform user study to verify the interestingness and representativeness of mine patterns. The subjective experiment is conducted by evaluating life patterns with different temporal conditions and temporal granularities; and conditional life patterns.

A. Settings

1) Data

Figure 10 shows the GPS devices we chose to collect data. They are comprised of stand-alone GPS receivers (Magellan Explorist 210/300, G-Rays 2 and QSTARZ) and GPS phones. All of them are set to receive GPS coordinates every two seconds. Using these devices, volunteers respectively log their life experience with GPS. The length of different

individuals' GPS logs varies according to the time they join the data collection. The earliest one records data of 16 months; the latest one has log of merely several weeks. All volunteers are suggested to switch on their devices as long as they travel outdoors. The data they collected covers 28 big cities in China and some cities in USA, South Korea, and Japan. The total distance of these GPS logs exceeds 50,000 KM.

Fig. 7 GPS devices used in the experiment

2) Parameter

In stay point detection, we set timeThreh to 30 minutes and distThreh to 200 meters. In other words, if an individual stays over 30 minutes within a distance of 200 meters, a stay point is detected.

In stay point clustering, we set NoP parameter to be 4 and disThre to be 0.15K.M. That is, if there are at least 4 stay points within 0.15km of an already clustered stay point, they will be added to the cluster.

These stay pint detection and clustering parameters enable us to find out each individual's significant places, such as restaurant, home, and shopping mall, etc., while ignoring the geographic regions without semantic meaning, like the places where people wait for traffic lights or meet congestion.

B. Objective Experiments

One major task of mining individual life pattern is to predict future life trends based on patterns mined from previous data. To evaluate the *predictability* of patterns mined by LP-Mine, we conduct following objective experiment:

- For each volunteer o, the life sequences dataset D^s is divided into two parts: D^{s1} contains sequences corresponding to the 1st, 3rd, 5th … days, D^{s2} contains sequences corresponding to the 2nd, 4th, 6th … days. Here we assign no temporal condition and the temporal granularity is set to "day".
- For support threshold set from 0.1 to 0.8, step by 0.1, we mine all non-sequential and sequential life patterns from D^{s1}. The temporal-annotation is set on all timestamp/span dimensions.
- For Individual o's life pattern p with support s, we use D^{s2} to match p. Here we introduce a "match coefficient" m, which equals the percentage of sequences in D^{s2} containing p. And the *predictability* of p is defined as:

$$pred(p) = 1 - \frac{|s-m|}{\max(s,m)} \quad (12)$$

- For each individual the predictability of all his/her patterns is defined as:

$$pred(o) = avg(pred(p)) \quad (13)$$

And the total predictability of all volunteers' life pattern is defined as:

$$pred = \sum weight(o) \cdot pred(o) \quad (14)$$

Here $weight(o)$ is assigned to o based on the number of data collecting days (different volunteers in our experiment have different data collecting days) and $\sum weigh(o) = 1$.

Figure 8 plots the predictability of all individual life patterns as the function of support threshold. For both sequential and non-sequential life patterns, the predictability grows with support threshold. This result is intuitive, because higher support corresponds to more general life style. Even when the support threshold equals 0.1, the predictability is also relatively high (0.73 for sequential patterns). To conclude, we claim that life patterns mined by LP-Mine can be reasonably employed to predict future life trends.

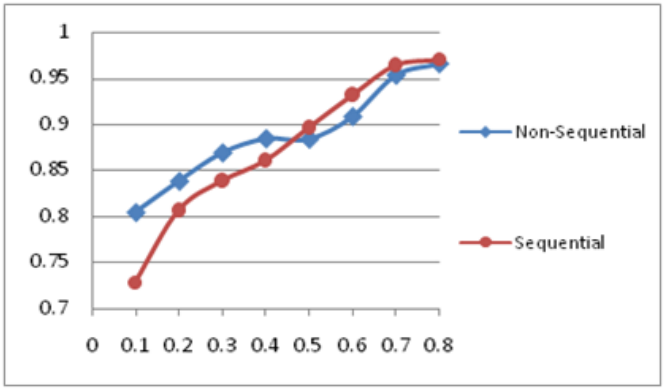


Fig. 8 Predictability of patterns as function of support threshold

C. Subjective Experiments

In subjective experiments, we visualize the life pattern by injecting them into a web application based on live search map called "*GeoLife*", thus we can conduct user study to collect user's judgement on the mined patterns.

From all volunteers, we choose 5 of them with longest GPS logs. We report the aggregate result (mean) of all users' study. We conduct three subjective experiments, separately on temporal condition, temporal granularity and conditional life pattern.

Temporal Condition: On temporal condition experiment, we set the temporal granularity to "day" and separately mine life patterns of all days, workdays and holidays. For each isolated pattern, it's hard for the individual to judge how interesting it is (the *interesting* measure), or how it represents general life style (the *representative* measure). Thus, on each temporal condition, we sort all patterns in descendant order of their supports. By observing the ranking of different patterns, the user can judge how interesting or representative these patterns totally represent. Taking one user (with longest GPS log, whom we use as example in this section) for instance, the pattern "*visit the girlfriend's house*" has support merely 0.12 on workdays, so it ranks low in workday's pattern sequence.

However, this pattern has support 0.72 on holidays, so it ranks highest on holiday's pattern sequence. The pattern "*visit the company*" has contrary result. The user can thus judge that the patterns of holidays are more interesting while less representative and the patterns of workdays are less interesting but more representative. In other words, interesting and representative are used to measure the method of setting temporal condition, instead of isolated patterns. The users are required to grade each condition's sequence of patterns on both measures with a scale of 0-5. The larger the grade, the more interesting or representative the individual judges it.

The experimental results were analysed by a one way ANOVA with the grading as the dependent variable, the task (the 3 types of temporal condition) as independent variable. It was found that the tasks ($F = 12.8, p < 0.007$ for interesting measure and $F = 10.7, p < 0.007$ for representative measure) significantly affected the grading. Figure 9 depicts the mean interesting and representative measure on different temporal conditions.

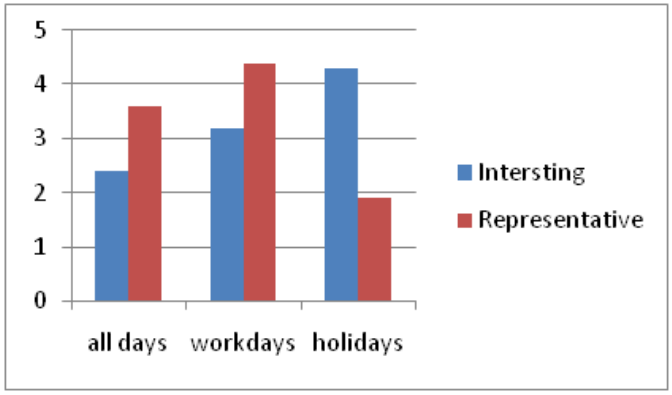


Fig. 9 Measurements on different temporal conditions

For the exemplary user, when condition is "all days" or "workdays", the patterns are mostly trivial, about home, company, etc., but when it comes to "holidays", we extract the most visited shopping mall, cinema, park etc. We also find the percentage of weekends when the individual takes overtime. The experiment on temporal condition accords intuition, revealing that more valuable life patterns tend to be found with special temporal condition.

Temporal Granularity: On temporal granularity experiment, we set no special temporal condition and separately construct life sequence dataset with temporal granularity "day", "week" and "month". Likewise, the individuals judge the interesting and representative measure of mined patterns. We also use ANOVA test to confirm that different granularities influence the result. Figure 10 plots the average grading.

When granularity is set to "week", we mined a lot of weekly repetitious life styles which cannot be discovered when granularity is "day". Like the weekly sports-taking place and time; the weekly regularity of going to a digital products market, etc. When granularity is "month", some patterns about visiting a friend and taking a short tour are discovered. The experiment on temporal granularity accords our discussion in Section III that life patterns with different granularities carry different semantics.

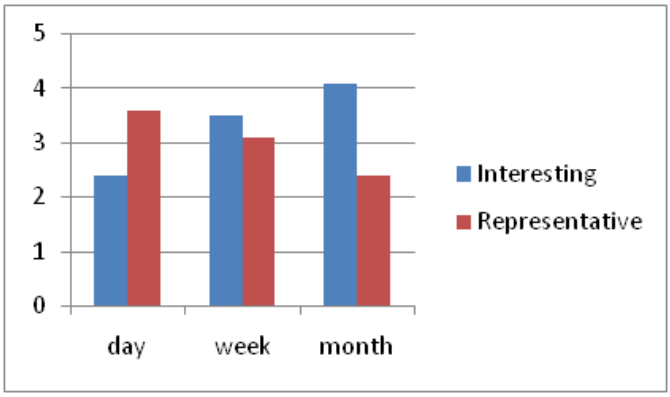


Fig. 10 Measurements on different temporal granularities

Conditional Life Pattern: We set temporal granularity to "day" and set three types of conditions: (1) not visiting the most frequent place; (2) visiting the second frequent place; (3) visiting the second frequent place while not visiting the most frequent place. The individuals grade the interesting and representative measure of patterns on each condition. Also ANOVA test certifies that different conditions affect result significantly. Figure 11 plots the average grading.

For the exemplary user, the most frequent place is "company"; the second is "girlfriends' home". On condition 1, most patterns contain "visiting the girlfriends' home"; on condition 2, there are a lot of trivial life patterns about home and companies. When it comes to condition 3, a lot of interesting patterns are mined, with regard to the places and time he tends to go out with his girlfriend.

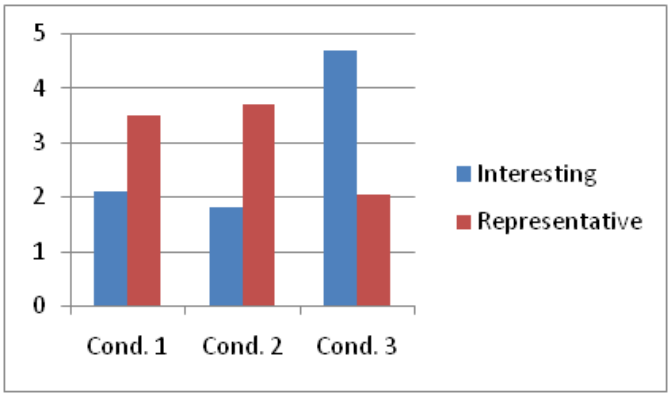


Fig. 11 Measurements on different conditions

D. Discussion

In the experiment, we discover a lot of trivial or uninteresting life patterns about home, working place, etc. This should be attributed to the derivation from "support" framework, thus patterns with high support are naturally discovered.

However, the paradox lies in that, for life pattern, the more frequent it is, the more trivial, or valueless it tends to be. Because most frequent life styles are trivial. Thus the utility of

LP-Mine would be limited. In the "conditional pattern" experiment, we personally design specific condition to retrieve interesting patterns. However, a better interestingness measure to substitute "support" and enhanced algorithm to automatically discover interesting patterns should be expected.

What is more, the temporal-annotated patterns are especially hard to be utilized because they typically contain timestamp or timespan interval with very large length. Although we include their mining in LP-Mine framework, experiments on them are not successfully conducted because mined temporal-annotated patterns are most hard to understand. Better formalization and methods to effectively extract temporal life knowledge is also a promising direction.

In the meantime, we find an urge of privacy preservation in the context of individual life pattern mining, since we mined a lot of private patterns in the experiment. A properly designed protocol to ensure personal privacy is required when the individual provides personal GPS data.

VII. CONCLUSIONS

In this paper, we extend the notion of frequent pattern into the context of GPS data; we propose a novel definition of life pattern; we present LP normal form to formalize the definition of individual life patterns; we propose LP-Mine, an abstraction-and-mining framework to effectively retrieve life patterns from GPS data.

This paper lies down a solid foundation for future works towards several directions: refinement of LP-Mine Framework; individual life knowledge mining from GPS data; multiple users' life knowledge mining from GPS data and life pattern, life pattern based application, etc.

- ***Refining the LP-Mine Framework***. As discussed in the experiment section, LP-Mine framework generates a lot of trivial or uninteresting life patterns and temporal-annotated patterns tend to be useless. We aims at investigating better measure instead of "support" to evaluate the value of life pattern. We also aim at refining both the framework and its implementation so as to enhance its utility. We shall also investigate privacy preservation techniques with individual life pattern mining system.
- ***Mining individual life knowledge from GPS data.*** The introduction of life pattern mining from GPS data in this paper may also be extended to other types of individual life knowledge mining, like "*outlier detection*", "*classification*", etc. Typically, life outlier detection is the counter part of pattern mining, which retrieves irregularity life activity. This irregularity generally corresponds to significant life change or events, so their detection is quite useful.
- ***Mining multiple users' life knowledge from GPS data.*** Although we focus on individual life pattern and associate rules, the formalization of life pattern and associate rules can also be extended to their counterpart of multiple user's. In the meantime, the data processing technique and mining technique can also be extended to mining multiple users' life pattern and associate rules.
- ***Life pattern based application.*** Individual life pattern and associate rule can be injected into a manifold of applications, including computer-aid blogging system, personal schedule system, route recommending, etc. What is more, friend recommendation system may be built on collecting multiple users' life patters.

REFERENCES

[1] J.-G. Lee, J. Han, X. Li, and H. Gonzalez, "Traclass: Trajectory classification using hierarchical region-based and trajectory-based clustering," in *Proc. of VLDB*, 2008.

[2] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. of SIGMOD*, 2007, pp. 593–604.

[3] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. of KDD*, 2007, pp. 330–339.

[4] H. Cao, N. Mamoulis, and D. W. Cheung, "Mining frequent spatiotemporal sequential patterns," in *Proc. of ICDM*, 2005, pp. 82–89.

[5] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Proc. of ICDE*, 2008, pp. 140–149.

[6] J. KRUMM and E. HORVITZ, "Predestination: Inferring destinations from partial trajectories," in *Proc. of UBICOMP*, 2006, pp. 243–260.

[7] F. Giannotti, M. Nanni, and D. Pedreschi, "Efficient mining of sequences with temporal annotations," in *Proc. of SIAM*, 2006, pp. 346–357.

[8] R. Agrawal, T. Imieliski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. of SIGMOD*, 1993, pp. 207–216.

[9] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. of ICDE*, 1995, pp. 3–14.

[10] J. Wang, J. Han, and J. Pei, "Closet+: searching for the best strategies for mining frequent closed itemsets," in *Proc. of SIGKDD*, 2003, pp. 236–245.

[11] X. Yan, J. Han and R. Afshar, "CloSpan: mining closed sequential patterns in large datasets," in *Proc. Of SDM, 2003*, pp 166–177.

[12] J. KRUMM and E. HORVITZ, "Predestination: Where do you want to go today?" *IEEE Computer Magazine*, vol. 40, no. 4, pp. 105–107, 2007.

[13] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Building personal maps from gps data," in *Proc. of IJCAI MOO05*, 2005, pp. 249–265.

[14] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," in *Proc. of the* National Conference on Artificial Intelligence. ACM Press, 2004, pp. 348–353.

[15] D. J. Patterson, L. Liao, D. Fox, and H. Kautz, "Inferring high-level behavior from low-level sensors," in *Proc. of UBICOMP*, 2003, pp. 73–89.

[16] J. KRUMM and E. HORVITZ, "Inferring motion and location from wi-fi signal strengths," in *Proc. of Mobiquitous*, 2004, pp. 4–13.

[17] T. Sohn, A. Varshavsky, A. LaMarca, and Y. Chen, "Mobility detection using everyday gsm traces," in *Proc. of UBICOMP*, 2006, pp. 212–224.

[18] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location hisotry," in *Proc. of GIS*, 2008 to appear.

[19] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," Data Mining and Knowledge Discovery, vol. 14, no. 1, 2007.

[20] P. Agarwal, S. Har-Peled, and K. Varadarajan, "Geometric approximation via corsets", in J. Goodman, J. Pach, and E. Welzl (Eds), Combinatorial and Computational Geometry, Cambridge University Press, 2005.

[21] P. Kumar, J. S. B. Mitchell, and E. A. Yildirim. "Approximate minimum enclosing balls in high dimensions using corsets", ACM J. Exp.Algorithm.