Mining Significant Time Intervals for Relationship Detection

Zhenhui Li Cindy Xide Lin Bolin Ding Jiawei Han

University of Illinois at Urbana-Champaign, Illinois, US

Abstract. Spatio-temporal data collected from GPS have become an important resource to study the relationships of moving objects. While previous studies focus on mining objects being together for a long time, discovering real-world relationships, such as friends or colleagues in human trajectory data, is a fundamentally different challenge. For example, it is possible that two individuals are friends but do not spend a lot of time being together every day. However, spending just one or two hours together at a location away from work on a Saturday night could be a strong indicator of friend relationship.
Based on the above observations, in this paper we aim to analyze and detect semantically meaningful relationships in a supervised way. That is, with an interested relationship in mind, a user can label some object pairs with and without such relationship. From labeled pairs, we will learn what time intervals are the most important ones in order to characterize this relationship. These significant time intervals, namely T-Motifs, are then used to discover relationships hidden in the unlabeled moving object pairs. While the search for T-Motifs could be time-consuming, we design two speed-up strategies to efficiently extract T-Motifs. We use both real and synthetic datasets to demonstrate the effectiveness and efficiency of our method.

1 Introduction

With the increasing popularity of GPS devices, the tracking of moving objects in general has become a reality. As a result, a vast amount of trajectory data is being collected and analyzed. Based on the temporal meeting pattern of objects, one of the most important and interesting problems in trajectory data analysis is *relationship detection*.

Previous studies of moving object relationships have been constrained to detecting moving object clusters. Studies such as flock [13], moving cluster [11], convoy [10], and swarm [16] focus on the discovery of a group of objects that move together. All these studies take the entire trajectory as a sequence of time points, and treat every time point or time interval *equally*. Therefore, the longer two moving objects are together, the better they are in terms of forming a cluster. However, all of these studies suffer from several drawbacks. On one hand, clusters discovered in this way usually do not carry any semantical meaning, such as friends, colleagues and families which naturally exist in human trajectory data. On the other hand, object pairs with certain relationship may not

necessarily meet more often than the other pairs, hence it leads to the failure of aforementioned methods in detecting such relationship. Considering the following example.

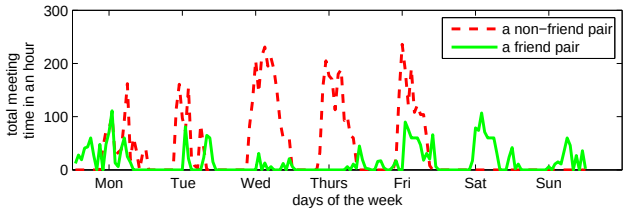


Fig. 1. Meeting frequency for a friend and a non-friend pair

Example 1. Reality Mining project[1] collected 94 human trajectories of the 2004-2005 academic year and conducted a survey about their friendship to each other. In Figure 1, we plot the meeting frequencies for one friend pair and one non-friend pair. Comparing two frequency curves, the one has overall higher meeting frequency is the non-friend pair. Thus, longer overall meeting time does not necessarily indicate friend relationship. In addition, we observe that the friend pair shows significantly higher meeting frequency on weekends, which indicates that, in the friend relationship case, not every time point has equal importance. In other words, since two people who meet more frequently on weekends are more likely to be friends, the weekend time interval are considered more discriminative and should play a more important role in the friend relationship detection task.

The above example reveals an important problem when analyzing relationship for moving objects: some time intervals are more discriminative than the others for a particular relationship. In fact, besides friend relationship, many relationships have their unique temporal patterns. For example, if we want to examine whether or not two people are colleagues, daytime on weekdays becomes the discriminative time intervals. If two people are family members, they often gather on holidays. Therefore, to detect semantically meaningful relationships in moving objects, we cannot treat all time intervals equally, instead we need to learn from the data what time intervals are the most important ones to characterize a relationship.

Consequently, in this paper we aim to detect relationship for moving object in a *supervised* way. That is, given a set of labeled data consisting of positive pairs having such relationship and negative pairs not having such relationship, our job is to first find those discriminative time intervals, namely *T-Motifs*. Then, these T-Motifs are used as features to detect this relationship in the remaining unlabeled pairs. Consider the following example.

[1] http://reality.media.mit.edu/

Example 2. In Reality Mining dataset, [21:56 Wed., 23:08 Wed.] is a T-Motif for friend relationship, because 37.3% friend pairs have meeting frequency more than 12 minutes in this time interval whereas only 3.17% non-friend pairs have meeting frequency that could reach 12 minutes.

According to the above example, we can interpret T-Motif as a time interval for which the meeting frequency between positive and negative pairs can be well split by a frequency value. Hence we propose to use *information gain* to measure the significance of a time interval. We need to calculate the significance score for any time interval and pick those intervals with high scores as T-Motifs. So the main technical challenge remains in the computation of significance score for huge number of T-Motif candidates.

To efficiently handle the large number of T-Motifs candidates, we design two efficient speed-up techniques for our algorithm. The first speed-up strategy is based on the observation that two similar time intervals, such as $[S, T]$ and $[S, T + 1]$, should have similar significance scores. Therefore, we propose to use time-indexed meeting pairs, so that when shifting the ending time from T to $T + 1$, we only need to update pairs who meet at time $T + 1$ and at the same time maintain the sorted list for all the pairs. The second speed-up technique takes advantage of skewed data. That is, positive pairs are only a small portion of all pairs. Based on a property of information gain, we could reduce the time to find the best split point from $O(|D|)$ to $O(|D^+|)$, where $|D^+|$ is the number of positive pairs. This further speeds up the computation when the positive pairs are only a small portion of all pairs, which is indeed the case for our problems.

In summary, the contributions of our work are as follows. (1) Our work is the first to detect semantically meaningful relationships in moving objects in a supervised way. This is done by introducing the concept of T-Motifs to properly represent the temporal characteristics for a relationship. (2) Two speed-up techniques are proposed to efficiently discover the T-Motifs. (3) The effectiveness and efficiency of our methods are demonstrated on both real and synthetic datasets.

The rest of this paper is organized as follows. Section 2 depicts the general framework. Section 3 describes the basic algorithm to mine T-Motifs. In Section 4, we introduce the speed-up techniques. Section 5 shows experimental results with respect to effectiveness and efficiency. Related work is discussed in Section 6, and the paper concludes in Section 7.

2 Problem Analysis

In this paper, the time intervals are defined in a *relative* time frame instead of an absolute one. This is because the movements of objects such as human usually have strong spatio-temporal regularities [8][6][19]. Therefore for human movements, for instance, it is more informative use "week" as the relative time frame. By default, we take minute as the basic time unit and consider any time point in a weekly time window (see Figure 1). Hence the total number of time

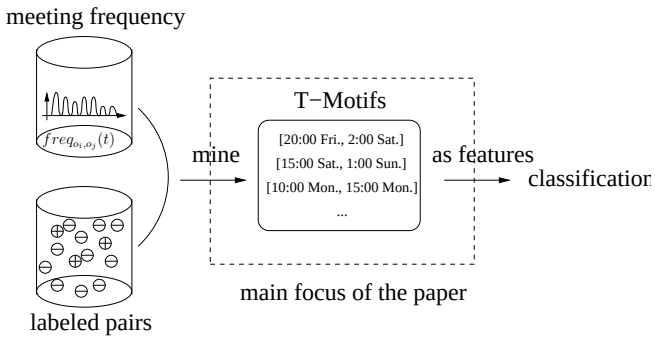


Fig. 2. Framework Overview

points is $P = 7$ (days) $\times$ 24 (hours) $\times$ 60 (minutes) $= 10080$. Any minute in the original absolute frame can be mapped to an integer from 1 to P. Similarly, a time interval $[S, T]$ is also defined in the relative time frame and should be understood in a cyclic order when $S > T$. The maximum length of a time interval is P by definition.

Let $O_{DB} = \{o_1, o_2, \ldots, o_n\}$ be the set of all moving objects. The meeting frequency between any two objects could be inferred from their movements. For any object pair (o_i, o_j) and time $t, 1 \leq t \leq P$, the meeting frequency $freq_{o_i,o_j}(t)$ is defined as the total number of times they meet at t in the relative time frame. There are various ways to determine whether two objects *meet* at one time. The most common way is to see whether the distance of their spatial locations are within certain distance threshold. Such distance threshold is based on the property of moving objects and the specific application. Another way could be using bluetooth to detect nearby objects, such as the dataset used in our experiment.

With a particular relationship in mind, a user can label some pairs of objects having or not having such relationship. We use D^+ and D^- to denote the set of positive and negative pairs, respectively. For example, we write $(o_i, o_j) \in D^+$ if objects o_i and o_j are labeled by the user to have such a relationship. Further, we use $D = D^+ \bigcup D^-$ to denote the set of all labeled pairs.

Figure 2 shows an overview of our framework. Given the meeting frequencies of the labeled pairs, a set of significant time intervals, namely T-Motifs, are extracted to capture the temporal characteristics of the relationship. Then, a classification model is built using T-Motif features of training data. The remaining unlabeled pairs can be classified using the learned classification model. In this framework, the most challenging part is how to find T-Motifs. In the following sections, we will present the definition of T-Motifs and an efficient method to extract them.

3 Finding T-Motifs

A T-Motif is a significant time interval to characterize the relationship. In this section, we will first describe how to calculate the significance score for a time interval and then analyze the basic algorithm to extract top-k T-Motifs.

3.1 Significance of Time Intervals

To examine the significance of a time interval $[S, T]$, we need to first calculate the meeting frequency for every pair in this time interval. Meeting frequency $freq_{o_i,o_j}(S, T)$ is the amount of time that o_i and o_j meet within the time interval $[S, T]$:

$$freq_{o_i,o_j}(S, T) = \sum_{t\in[S,T]} freq_{o_i,o_j}(t).$$

In addition, for any set of pairs A consisting of positive pairs A^+ and A^-, its *entropy* is

$$H(A) = -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha),$$

where $\alpha = \frac{|A^+|}{|A|}$ is the fraction of positive numbers.

Intuitively, a time interval $[S, T]$ is significant if a large portion of positive (or negative) pairs have higher meeting frequencies than most of the negative (or positive) pairs. Once we select a meeting frequency value v as the *split point*, the pairs in D having meeting frequency in $[S, T]$ no less than split point form the set $D^{S,T}_{\geq}(v)$ and the rest form $D^{S,T}_{<}(v)$:

$$D^{S,T}_{\geq}(v) = \{(o_i, o_j)|freq_{o_i,o_j}(S, T) \geq v\}, D^{S,T}_{<}(v) = \{(o_i, o_j)|freq_{o_i,o_j}(S, T) < v\}.$$

The *information gain* of $[S, T]$ at split point v is:

$$IG^{S,T}(v) = H(D) - \frac{|D^{S,T}_{\geq}(v)|}{|D|} H(D^{S,T}_{\geq}(v)) - \frac{|D^{S,T}_{<}(v)|}{|D|} H(D^{S,T}_{<}(v)).$$

The *significance score* of $[S, T]$ is the highest information gain that any split point can achieve:

$$G(S, T) = \max_v IG^{S,T}(v).$$

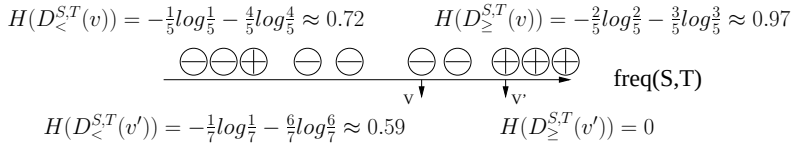


Fig. 3. An example for calculation of $G(S, T)$

This concept is illustrated in Figure 3.

Example 3. Suppose the labeled set D contains 4 positive pairs and 6 negative pairs. Figure 3 shows the meeting frequency of each pair in a time interval $[S, T]$. We compute $H(D) = -\frac{4}{10}log\frac{4}{10} - \frac{6}{10}log\frac{6}{10} \approx 0.97$. At split point v, the information gain of $[S, T]$ is $IG^{S,T}(v) = 0.97 - \frac{5}{10} \times 0.72 - \frac{5}{10} \times 0.97 \approx 0.125$. The highest information gain is achieved at split point v': $G(S, T) = IG^{S,T}(v') = 0.97 - \frac{7}{10} \times 0.59 - \frac{3}{10} \times 0 \approx 0.557$.

3.2 Overview of Basic Algorithm

Algorithm 1 Find T-Motifs

Input:
$freq$: meeting frequency for each pair;
D^+: positive pairs;
D^-: negative pairs.
Output: T-Motifs.
Algorithm:

```
1: D ← D+ ⋃ D−
2: for S ← 1 to P do
3:    for len ← δmin to δmax do
4:       T ← S + len − 1
5:       freq_arr ← {freq_{oi,oj}(S, T), ∀(oi, oj) ∈ D}
6:       Sort freq_arr
7:       for i ← 1 to |D| do
8:          v ← freq_arr(i)
9:          if IG(D, v) > best_IG then
10:            best_IG = IG(D, v)
11:      G(S, T) = best_IG
12: Return top-k non-overlapped time intervals
```

The basic algorithm is summarized in Algorithm 1. To find the T-Motifs, we first need to compute the significance score (i.e., information gain) for every time interval $[S, T]$. But sometimes it is unnecessary to consider time intervals which are too short or too long, such as one-minute time interval or the intervals with maximum length such as $[1, P]$. So the algorithm has an option to limit the length of time interval to $[\delta_{min}, \delta_{max}]$, where δ_{min} and δ_{max} are specified by the user. Now, for a time interval $[S, T]$, to get the meeting frequency of each pair takes $O(|D|)$ time (Line 5 in Algorithm 1). In order to calculate the significance score, the pairs will be sorted first (Line 6 in Algorithm 1). The sorting takes $O(|D| \log |D|)$ time. Taking each meeting frequency as split point v, the information gain $IG^{S,T}(v)$ can be calculated (Line 7-10 in Algorithm 1). The time complexity for this step is $O(|D|)$. And finally the maximal information gain value is set as the significance score for interval $[S, T]$. With the significance scores for all time intervals, we pick the top-k non-overlapped time intervals as

T-Motifs. This procedure is similar to the selection of discriminative patterns in [3][17].

From Algorithm 1, we can see that the number of all time intervals is $O(P^2)$ in the worst case. And for each time interval $[S, T]$, it takes $O(|D| \log |D|)$ to compute the significance score. So the overall time complexity is $O(P^2|D| \log |D|)$.

4 Speed Up the Searh for T-Motifs

In this section, we propose two accelerating techniques to our basic algorithm. The first one is to build a time-indexed data structure, which allows us to quickly retrieve the pairs that meet at a certain time point T, and locally adjust the order of all pairs based on the changes in meeting frequencies. The second speed-up technique is based on an important property of information gain, which greatly reduces the number of split points one needs to examine for each time interval in order to compute its significance score.

4.1 Time-indexed Meeting Pairs

In Algorithm 1, for a time interval $[S, T]$, we need to compute the meeting frequency for every pair (Line 5), which takes $O(|D|)$ time. However, it may be unnecessary to update *every* pair when expanding time interval from $[S, T]$ to $[S, T+1]$, since in the real data only a limited number of pairs meet at time $T+1$. For any time point t, to retrieve the pairs that meet at t, we use a time-based list $T\_list(t)$ to record those pairs,

$$T\_list(t) = \{(o_i, o_j) | freq_{o_i,o_j}(t) \neq 0\}.$$

With this data structure, Line 5 in Algorithm 1 can be replaced by retrieving every pair stored in $T\_list(t)$ and just update frequencies for those pairs. Even though T_list takes $\Omega(P \cdot d)$ additional memory, where d is the average number of meeting pairs per time point, it helps the updating step reduce its time complexity from $O(|D|)$ to $O(d)$. In real scenarios, as shown in our experiments (Section 5.3), d is usually much smaller than $|D|$

After updating frequencies, all the pairs need to be sorted according to their frequencies (Line 6 in Algorithm 1), which takes $O(|D| \log |D|)$ time. But when expanding T to $T + 1$, only a few pairs update their frequencies. Therefore, instead of doing sort all over again, we can update the sorted list with a small number of adjustments.

Take Figure 4 for example. All the pairs are sorted in ascending order when ending time is T. When a pair increases its meeting frequency from 6 to 16 for time interval $[S, T + 1]$, it switches its position with the one on the right repeatedly, until it reaches the right-most position or the value on the right is larger than 16.

To update the position for one pair, it takes $O(|D|)$ in the worst case. In total, it takes $O(d|D|)$ to adjust the positions in the sorted list for all the updated

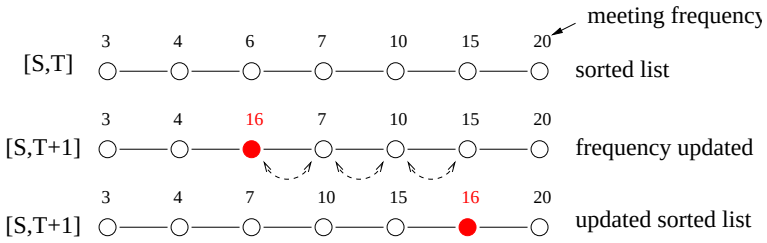


Fig. 4. Updating sorted list

pairs in T_list. Theoretically, this sorting strategy is no better than fast sort ($O(|D|\log|D|)$) because d may not be smaller than $\log|D|$. However, it takes much less time in practice since one pair will not increase its meeting frequency drastically by expanding ending time from T to $T+1$. We will verify this in experiment.

4.2 Finding the Best Split Point

Given the order of all pairs, it takes $O(|D|)$ time to consider each pair as a split point and calculate corresponding significance score. We next prove that it suffices to enumerate the values of all positive pairs, which takes $O(|D^+|)$ time. We observe that, in most real scenarios, the data have an important property: the number of pairs having the labeled relationship only takes a small portion of all the pairs (i.e., $|D^+| \ll |D|$).

For a split point v in time interval $[S,T]$, let $p(v)$ and $q(v)$ be the fractions of positive and negative pairs whose meeting frequencies are no less than v, respectively:

$$p(v) = \frac{|D^+ \bigcap D_{\geq}^{S,T}(v)|}{|D^+|},\ q(v) = \frac{|D^- \bigcap D_{\geq}^{S,T}(v)|}{|D^-|}. \tag{1}$$

Given a pair $(p(v), q(v))$, we can write the information gain $IG(v)$ as a function of p and q:

$$IG(v) = IG(p(v), q(v)).$$

To introduce our second speed-up technique, we will find the following general property of information gain useful:

Lemma 1. *Given a pair of probabilities (p,q) as defined in (1), we have the following two properties of information gain:*

1. *if $p > q$, then $\frac{\partial IG}{\partial p} > 0, \frac{\partial IG}{\partial q} < 0$,*
2. *if $p < q$, then $\frac{\partial IG}{\partial p} < 0, \frac{\partial IG}{\partial q} > 0$.*

Basically, the above lemma states that if the frequency difference of positive pairs and negative pairs increases, then the split point v becomes more significant. In fact, in addition to information gain, it can be proven that many other popular statistical measures, such as the G-test score, also satisfy this good property. Interested readers are referred to [23] and the reference therein for the proof and more discussion of the lemma.

In the context of our work, Lemma 1 could be interpreted as follow.

Corollary 1. *For any two split points v_1 and v_2 and a time interval $[S, T]$,*

1. *if $p(v_1) > q(v_1)$, $p(v_2) \geq p(v_1)$ and $q(v_2) \leq q(v_1)$ where the two equalities do not hold simultaneously, then $IG(v_2) > IG(v_1)$,*
2. *if $p(v_1) < q(v_1)$, $p(v_2) \leq p(v_1)$ and $q(v_2) \geq q(v_1)$ where the two equalities do not hold simultaneously, then $IG(v_2) > IG(v_1)$.*

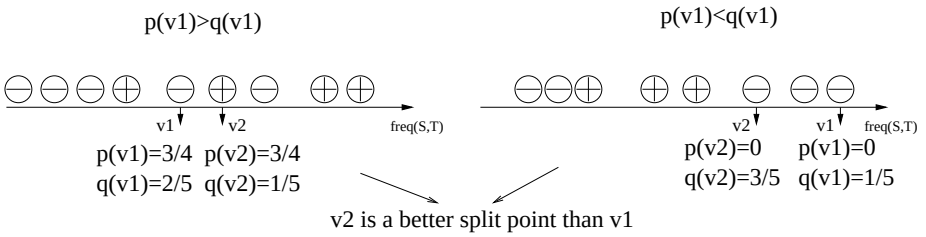


Fig. 5. Illustration for Corollary 1

The left subfigure of Figure 5 illustrates the first case in Corollary 1, i.e., $p(v_1) > q(v_1)$. As we can see, v_1 takes a negative pair as a split point. If we select v_2 as the split point, there is one less negative pair on the right side (i.e., $q(v_2) < q(v_1)$) but the number of positive pairs on the right side remains the same (i.e., $p(v_2) = p(v_1)$). Since the difference between positive pairs and negative pairs on the right side increases, $IG(v_2) > IG(v_1)$. In practice, we can observe the following two facts in the real data: (1) $|D^-| \gg |D^+|$ and (2) a considerable portion of negative pairs have very low or zero meeting frequency for any given time interval $[S, T]$. Therefore, for any time interval $[S, T]$, we can always assume $p(v) > q(v)$, where $0 < v \leq \max_{(o_i,o_j)\in D^+}\{freq_{o_i,o_j}(S, T)\}$. With this assumption, we can skip any negative pair whose meeting frequency is no larger than the maximum meeting frequency among all positive pairs.

Therefore, in addition to the positive pairs, we only need to examine those negative pairs which are on the right of the rightmost positive pair, as shown in the right subfigure of Figure 5. In fact, among all of these negative pairs, examining the leftmost one suffices. This is because in this case we have $p(v_1) = 0 < q(v_1)$. Therefore, by further shifting the split point to the left from v_1 to v_2, we always have $p(v_2) = p(v_1) = 0$ and $q(v_2) > q(v_1)$, thus $IG(v_2) > IG(v_1)$ by Corollary 1.

We summarize our second speed-up technique as the following theorem.

Theorem 1. *For a time interval $[S,T]$, assume the maximum meeting frequency of all positive pairs is v^*, then the best split point v must take one of following values:*

1. *the value of one positive pair: $v \in \{freq_{o_i,o_j}(S,T) : (o_i,o_j) \in D^+\}$,*
2. *the smallest value for the negative pairs that is larger than the maximum meeting frequency of all positive pairs:*

$$v = \min_{(o_i,o_j)\in D^-} \{freq_{o_i,o_j}(S,T) : freq_{o_i,o_j}(S,T) > v^*\}.$$

Therefore, for a time interval $[S,T]$, it takes $O(|D^+|)$ to compute significance score $G(S,T)$. Since $|D^+| \ll |D|$ in our problem, this saves a lot of time comparing with original time complexity $O(|D|)$.

5 Experiment

Our experiments are carried on both real and synthetic datasets. All the algorithms are implemented in C++, and all the experiments are carried out on a 2.13 GHz Intel Core machine with 4GB memory, running Linux with version 2.6.18 and gcc 4.1.2.

5.1 Dataset Description

To evaluate the effectiveness of our method, we use the Reality Mining dataset[2]. The Reality Mining project was conducted from 2004-2005 at MIT Media Laboratory. The study followed 94 subjects using mobile phones. We filtered the subjects with missing information and 86 subjects are left. The proximity between subjects can be inferred from repeated Bluetooth scans. When a Bluetooth device conducts a discovery scan, other Bluetooth devices within a range of 5-10 meters respond with their user-defined names. Therefore, instead of using the trajectory data, we take the Bluetooth data directly to generate the meeting frequencies for any two subjects. Although 86 subjects should form 3655 pairs, there are 1856 pairs which have zero meeting frequency. After filtering those pairs, our dataset has 1799 pairs in total.

We study two relationships on this dataset.

- **Friend relationship.** Subjects were asked about their friend relationship to the other individuals in the study. The survey question was "Is this person a part of your close circle of friends?" According to the survey, there are 22 reciprocal friend pairs, 39 non-reciprocal friend pairs and 1718 reciprocal non-friend pairs. In the survey, subjects were also asked about their physical proximity with the other individuals. In this analysis, we only use the pairs who have mutually reported some proximity. By doing so, we filter out those

[2] http://reality.media.mit.edu/

pairs with low meeting frequencies. The reason of doing this is because it is more trivial to achieve high accuracy if we include those pairs with few interactions. Similar pre-processing step was conducted in work [7] to study friend relationship. In the remaining pairs, we take reciprocal and non-reciprocal friend pairs as positive pairs (i.e., $|D^+| = 59$) and the non-friend pairs as negative ones (i.e., $|D^-| = 441$).

– **Colleague relationship.** One subject could belong to one of the affiliations such as media lab graduate student, media lab staff, professor, Sloan business school, etc.. To study colleague relationship, we take all the pairs belong to Sloan business school as the positive pairs and the remaining ones as the negative pairs. There are 218 positive pairs (i.e., $|D^+| = 218$) and 1561 (i.e., $|D^-| = 1561$) negative pairs in this case.

5.2 Discovery of T-Motifs

In this section, we will show the T-Motifs for friend relationship and colleague relationship separately.

T-Motif $[S, T]$	best split point v	$\frac{\lvert D_{\geq}^{S,T}(v) \cap D^+ \rvert}{\lvert D^+ \rvert}$	$\frac{\lvert D_{\geq}^{S,T}(v) \cap D^- \rvert}{\lvert D^- \rvert}$
[21:56 Wed., 23:08 Wed.]	12	0.372881	0.031746
[22:45 Tue., 23:39 Tue.]	55	0.305085	0.0181406
[19:07 Sat., 7:07 Sun.]	249	0.220339	0.00453515
[20:56 Tue., 22:44 Tue.]	1	0.508475	0.113379
[23:55 Tue., 1:42 Wed.]	10	0.355932	0.0453515
[23:22 Wed., 3:43 Thurs.]	53	0.220339	0.00680272
[7:08 Sun., 16:49 Sun.]	53	0.40678	0.0770975
[1:20 Fri., 5:12 Fri.]	12	0.20339	0.00680272
[21:52 Mon., 9:00 Tue.]	11	0.644068	0.240363
[18:12 Sun., 20:01 Sun.]	3	0.389831	0.0793651

Table 1. Top-10 T-Motifs for friend relationship

Table 1 shows the top-10 T-Motifs mined for friend relationship. Among all time intervals, [21:56 Wed., 23:08 Wed.] plays the most important role, as 37.3% friends have meeting frequency more than 12 minutes whereas only 3.17% non-friends can exceed 12-minutes meeting frequency. As one can see, the interactions at night are more discriminative for friend relationship in general, with exceptions during the daytime on weekends, such as [7:08 Sun., 16:49 Sun.].

Table 2 shows the top-10 T-Motifs for the colleague relationship. Interestingly, the colleague pairs (students from Sloan business school) usually have high meeting frequencies during the morning, especially on Tuesdays and Thursdays. It may suggest that these students have classes on Tuesday and Thursday mornings. Comparing to the friend relationship, it is obvious that colleagues have quite different temporal interaction patterns. T-Motifs provie us an insight in the uniqueness of each relationship.

T-Motif $[S,T]$	best split point v	$\frac{\lvert D_{\geq}^{S,T}(v)\cap D^{+}\rvert}{\lvert D^{+}\rvert}$	$\frac{\lvert D_{\geq}^{S,T}(v)\cap D^{-}\rvert}{\lvert D^{-}\rvert}$
[9:20 Thurs., 10:30 Thurs.]	7	0.7201	0.0557
[9:42 Tue., 10:35 Tue.]	3	0.7431	0.0749
[10:36 Tue., 11:34 Tue.]	56	0.6376	0.0384
[10:34 Thurs., 11:04 Thurs.]	31	0.6055	0.0358
[11:05 Thurs., 11:40 Thurs.]	31	0.6146	0.0589
[7:31 Tue., 8:44 Tue.]	1	0.4449	0.0109
[21:16 Thurs., 9:10 Fri.]	7	0.6376	0.0723
[8:02 Thurs., 8:49 Thurs.]	2	0.4220	0.0128
[8:45 Tue., 9:19 Tue.]	22	0.3853	0.0070
[5:32 Wed., 10:28 Wed.]	2	0.5917	0.0749

Table 2. Top-10 T-Motifs for colleague relationship

5.3 Efficiency Study

In this section, we analyze the scalability issue w.r.t. different data sizes and parameter settings. We compare the T-Motif mining baseline method as shown in Algorithm 1 (denoted as baseline) with the one with speed-up techniques (denoted as speedup). We first present the comparison results on the friend relationship data. By default, we will use all the pairs in D to find T-Motifs and set $\delta_{min} = 1$ and $\delta_{max} = 6 \times 60$.

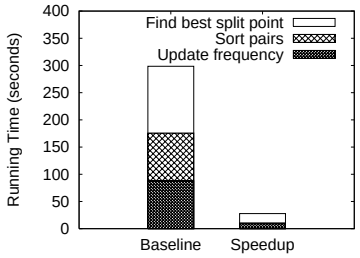


Fig. 6. Time spent on each step (friend relationship)

Figure 6 shows the time spent on each step when computing the significance scores for time intervals. In baseline method, to compute $G(S,T)$, there are three steps: updating meeting frequency with the time complexity $O(|D|)$, sorting all pairs with the time complexity $O(|D|log|D|)$ and finding the best split point with the time complexity $O(|D|)$. As shown in Figure 6, all the steps take roughly the same time. Compared to baseline, speedup compresses updating and sorting of the meeting frequencies into one step. As we mentioned in Section 4.1, even though this step takes $O(d|D|)$ time in theory, it is actually much faster in practice. In particular, updating and sorting by speedup together take 10.15 seconds whereas they take 175 seconds for baseline. The reason is illustrated in Figure 7. For many

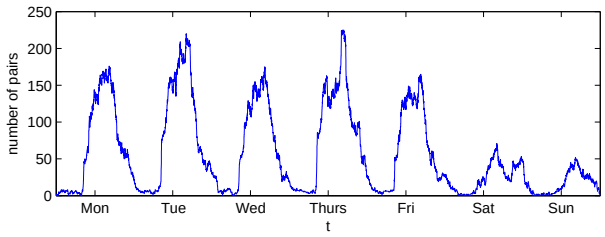
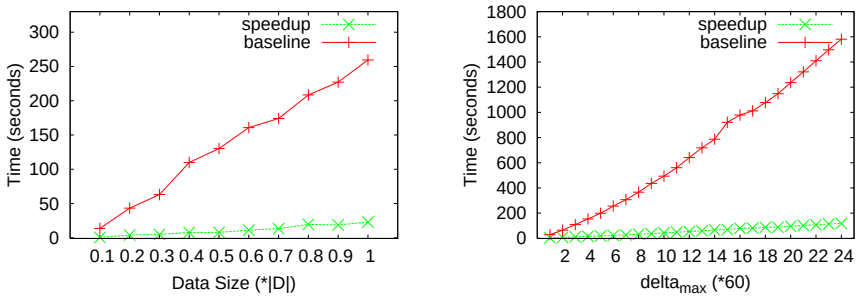


Fig. 7. Number of pairs meeting at each time point (friend relationship)



(a) Running time w.r.t data size (b) Running time w.r.t δ_{max}

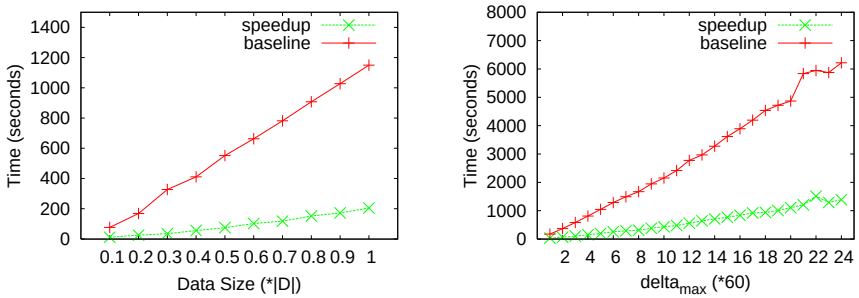
Fig. 8. Running time on friend relationship

time points, especially the ones at mid-night, very few pairs meet. On average, there are only 56.86 pairs meeting at each time point. Therefore, it is unnecessary to update the frequency for every pair and sort the frequencies all over again. Besides, the second speed-up technique reduces the best split point step from 299 seconds to 28 seconds since we only need to enumerate positive pairs as the split points.

Now we randomly select $p\%$ of the pairs from the entire dataset as the training samples and apply Algorithm 1. $p\%$ is enumerated from 10% to 100% with an increment of 10% in each trial. Figure 8(a) shows the running time w.r.t different data sizes. It is obvious that speedup techniques make the T-Motifs mining process much faster. The difference between speedup and baseline becomes bigger as the data size increases. When applying to the whole dataset, baseline takes 260 seconds whereas speedup only takes 23 seconds.

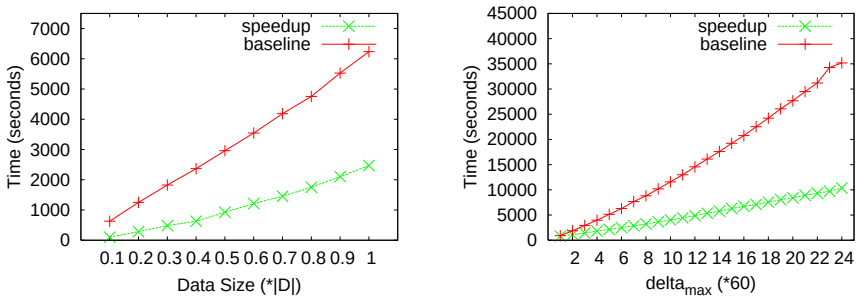
In Algorithm 1, we use δ_{min} and δ_{max} to limit the length of time intervals. When $\delta_{max} - \delta_{min}$ increases, the search space for T-Motifs also increases. By setting $\delta_{min} = 1$, we increase δ_{max} by hour, until it reaches 24 hours. The running times for baseline and speedup are plotted in Figure 8(b). Again, speedup is significantly faster than baseline, especially when δ_{max} is large.

We also conduct the same experiments on the dataset of colleague relationship. Figure 9 shows the running times w.r.t. different parameters, from which we can see that speedup is much faster than baseline.



(a) Running time w.r.t data size (b) Running time w.r.t δ_{max}

Fig. 9. Running time on colleague relationship



(a) Running time w.r.t data size (b) Running time w.r.t δ_{max}

Fig. 10. Running time on synthetic dataset

Finally, we synthesize an even larger dataset based on the Reality Mining dataset. In our synthetic dataset, there are $1,000$ positive pairs and $10,000$ negative pairs. To generate a new positive pair, we randomly select one positive pair from the Reality Mining dataset and perturb its meeting frequency for each time point within 10% of the original value. We repeat this process to generate new negative pairs. We report the efficiency w.r.t. data size and δ_{max} in Figure 10. Compared to Figure 8, the difference in running time between the two methods is getting bigger in larger dataset.

5.4 Relationship Detection using T-Motifs

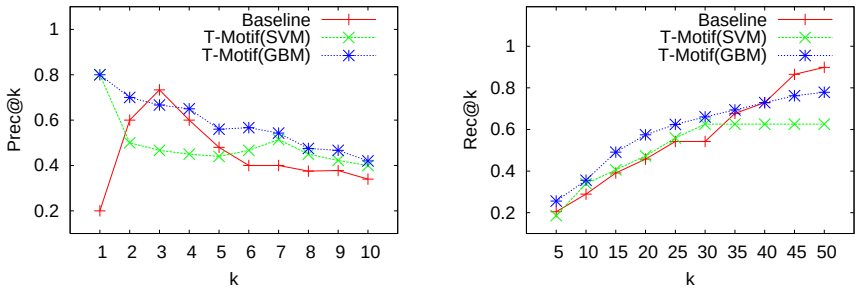
Next we use the extracted T-Motifs to find the interested relationship in unlabeled data. In this experiment, we set $\delta_{min} = 1$ and $\delta_{max} = 12 * 60$ to mine T-Motifs and use the top-20 T-Motifs. A classification model is built using labeled pairs as training data and the meeting frequency within each T-Motif as the feature. We report the classification results using Support Vector Machine (SVM) and Gradient Boosting Machine (GBM) as learning methods. For com-

parison, we set the baseline as directly counting the meeting frequency over the entire time frame, which is equal to $freq_{o_i,o_j}(1, P)$ for a pair (o_i, o_j).

Using the classification model, we will get a score for each test sample indicating the probability to be a positive pair. In the top-k ranked test samples S_k, we use *precision* to examine the ratio of true positives, and *recall* to measure how many pairs having such relationship are retrieved. Precision and recall are defined as:

$$Prec@k = \frac{|D^+_{test} \bigcap S_k|}{k}, Rec@k = \frac{|D^+_{test} \bigcap S_k|}{|D^+_{test}|},$$

where D^+_{test} is the set of positive pairs in the test set.

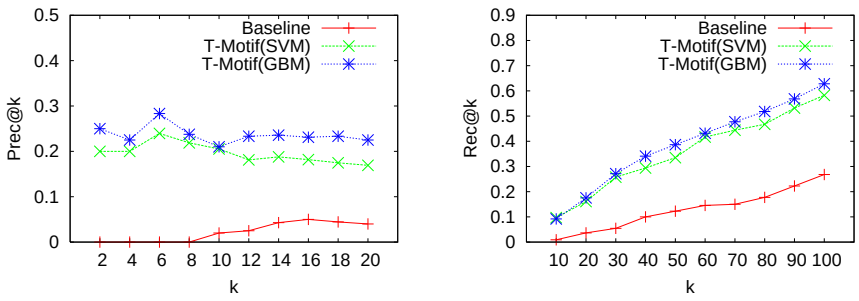


(a) Precision (b) Recall

Fig. 11. Effectiveness comparison on friend relationship

We use 5-fold cross validation on the friend relationship data. Figure 11 shows the precision and recall of T-Motif based methods comparing with that of baseline. We can see that Prec@1=0.8 for both T-Motif(SVM) and T-Motif(GBM). It means that, when using T-Motifs, 80% top-1 pairs are real friends. In contrast, Prec@1=0.2 for baseline method, which indicates that the pair that has the highest meeting frequency does not necessarily have the friend relationship. In terms of recall measure in Figure 11(b), the methods based on T-Motif have higher recall value when k is smaller than 30. It means that T-Motifs can promote friend pairs to higher ranks. But it is worth noting that the baseline can retrieve 89% friend pairs at top-50 ranked pairs. This suggests that friends generally meet more frequently.

For colleague relationship, we use 10-fold cross validation. As one can see in Figure 12, baseline method performs poorly on this dataset, as it barely retrieves any colleague pair in the top-20 pairs. Even in the top-100 pairs, as shown in Figure 12(b), baseline can only retrieve less than 30% colleague pairs. This indicates that the pairs meeting very often are not necessarily colleagues. As we have seen in Table 2, colleagues meet only at particular times. These patterns are well captured by T-Motifs, and we can retrieve 60% colleague pairs from top-100 pairs using T-Motifs.



(a) Precision (b) Recall

Fig. 12. Effectiveness comparison on colleague relationship

6 Related Work

Previous studies mainly focus on the discovery of one specific relationship - moving object clusters, such as flock [13], moving cluster [11], convoy [10], and swarm [16]. They try to find a group of objects that move together for k consecutive or non-consecutive times. All these work simply count the timestamps that objects are being together. Laube *et al.* [14][9] define several spatio-temporal patterns, including flock, leadership, convergence, and encounter. However, each pattern is defined and studied individually. These patterns cannot be generalized to detect any real-world relationship for moving objects.

Many methods have been proposed to measure the similarity between two trajectories, such as Dynamic Time Warping (DTW) [25], Longest Common Subsequences (LCSS) [20], Edit Distance on Real Sequence (EFR) [2], and Edit distance with Real Penalty (ERP) [1]. The geometric trajectory distance plays an important role in determining the similarity between two objects. However, two objects having some relationship may not necessarily have similar trajectories. There are also studies to measure the similarities between two human trajectories [15][22]. But the similarity is measured by the travel sequence, such as shopping $\rightarrow$ movie $\rightarrow$ restaurant. Such similarity does not consider the times that two objects are being close.

There are several interesting studies showing the potential of using mobile or positioning technologies to study the human social behavior, such as mobility regularity [8][6][19] and interactions [18][7][4]. Miklas *et al.* [18] and Eagle *et al.* [7] focus on the analysis of the relationships between physical network and social network. [18] finds that "friends meet more regularly and for longer duration whereas the strangers meet sporadically" and [7] shows that "friend dyads demonstrate distinctive temporal and spatial patterns in their physical proximity and calling patterns". A more recent work by Cranshaw *et al.* [4] develops a technique to infer friendship in a supervised way, which is the most related work to ours. They design a set of spatial-temporal features and build a classification model for friend relationship. Their temporal features, such as the number of co-locations in evening/weekends, are heuristically designed. Our work is a much more general approach to detect any relationship with any temporal patterns.

Our idea of making use of T-Motifs is also motivated by a recent work [24] on time series classification problem. Different from other time series classification methods [12][21][5], Ye *et al.* [24] use shapelets, which are time series subsequences that can maximally represent a class. The pruning rules developed in [24] try to avoid the expensive time cost to compute the distance between a shapelet and a time series. Our problem is different from typical time series classification because our time dimension is fixed to a relative time frame, such as a week, and it only takes $O(1)$ to calculate the meeting frequency for each object pair. Our speed-up techniques aim to save time for computing significance scores.

7 Conclusion

In this paper, we introduce a supervised framework to detect relationships for moving objects from their meeting patterns. In this framework, the concept of T-Motifs is proposed to capture the temporal characteristics for relationships. A T-Motif is a time interval which has high information gain with respect to meeting frequencies for labeled pairs. We develop two speed-up techniques to enumerate T-Motif candidates and calculate their significance scores. In the experiments with real-world datasets, the proposed method is both efficient and effective in discovering the relationship for moving objects. Extensions to make use of spatial features to better detect the relationships could be interesting themes for future research.

Acknowledgment

The work was supported in part by The Boeing Company, NSF IIS-1017362, NSF CNS-0931975, U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and by the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. L. Chen and R. T. Ng. On the marriage of lp-norms and edit distance. In *VLDB*, pages 792–803, 2004.
2. L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *SIGMOD Conference*, pages 491–502, 2005.
3. H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *ICDE*, pages 716–725, 2007.

4. J. Cranshaw, E. Toch, J. I. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *UbiComp*, pages 119–128, 2010.
5. H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *PVLDB*, 1(2):1542–1552, 2008.
6. N. Eagle and A. Pentland. Eigenbehaviors: identifying structure in routine. In *Behavioral Ecology and Sociobiology*, pages 1057–1066, 2009.
7. N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. In *Proceedings of the National Academy of Sciences*, pages 15274–15278, 2009.
8. M. C. González, C. A. H. R., and A.-L. Barabási. Understanding individual human mobility patterns. *Nature*, 453:779–782, 2008.
9. J. Gudmundsson, P. Laube, and T. Wolle. Movement patterns in spatio-temporal data. In *Encyclopedia of GIS*, pages 726–732, 2008.
10. H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of convoys in trajectory databases. *PVLDB*, 1(1):1068–1080, 2008.
11. P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *SSTD*, pages 364–381, 2005.
12. E. J. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov.*, 7(4):349–371, 2003.
13. P. Laube and S. Imfeld. Analyzing relative motion within groups of trackable moving point objects. In *GIScience*, pages 132–144, 2002.
14. P. Laube, M. J. van Kreveld, and S. Imfeld. Finding remo - detecting relative motion patterns in geospatial lifelines. In *Int. Symp. on Spatial Data Handling*, 2004.
15. Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *GIS*, page 34, 2008.
16. Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *PVLDB*, 3(1):723–734, 2010.
17. D. Lo, H. Cheng, J. Han, S.-C. Khoo, and C. Sun. Classification of software behaviors for failure detection: a discriminative pattern mining approach. In *KDD*, pages 557–566, 2009.
18. A. G. Miklas, K. K. Gollu, K. K. W. Chan, S. Saroiu, P. K. Gummadi, and E. de Lara. Exploiting social interactions in mobile systems. In *Ubicomp*, pages 409–428, 2007.
19. C. Song, Z. Qu, N. Blumm, and A. L. Barabasi. Limits of predictability in human mobility. In *Science*, pages 1018–1021, 2010.
20. M. Vlachos, D. Gunopulos, and G. Kollios. Discovering similar multidimensional trajectories. In *ICDE*, pages 673–684, 2002.
21. X. Xi, E. J. Keogh, C. R. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *ICML*, pages 1033–1040, 2006.
22. X. Xiao, Y. Zheng, Q. Luo, and X. Xie. Finding similar users using category-based location history. In *GIS*, pages 442–445, 2010.
23. X. Yan, H. Cheng, J. Han, and P. S. Yu. Mining significant graph patterns by leap search. In *SIGMOD Conference*, pages 433–444, 2008.
24. L. Ye and E. J. Keogh. Time series shapelets: a new primitive for data mining. In *KDD*, pages 947–956, 2009.
25. B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *ICDE*, pages 201–208, 1998.