Mining Social Network from Spatio-Temporal Events

Hady W. Lauw* Ee-Peng Lim* Teck-Tim Tan† Hwee-Hwa Pang‡

Abstract

Knowing patterns of relationship in a social network is very useful for law enforcement agencies to investigate collaborations among criminals, for businesses to exploit relationships to sell products, or for individuals who wish to network with others. After all, it is not just what you know, but also whom you know, that matters. However, finding out who is related to whom on a large scale is a complex problem. Asking every single individual would be impractical, given the huge number of individuals and the changing dynamics of relationships. Recent advancement in technology has allowed more data about activities of individuals to be collected. Such data may be mined to reveal associations between these individuals. Specifically, we focus on data having space and time elements, such as logs of people's movement over various locations or of their Internet activities at various cyber locations. Reasoning that individuals who are frequently found together are likely to be associated with each other, we mine from the data instances where several actors co-occur in space and time, presumably due to an underlying interaction. We call these spatio-temporal co-occurrences events, which we use to establish relationships between pairs of individuals. In this paper, we propose a model for constructing a social network from events, and provide an algorithm that mines these events from the data. Experiments on a real-life data tracking people's accesses to cyber locations have also yielded encouraging results.

Keywords

social network, spatio-temporal data mining, link analysis

1 Introduction

Social network describes a group of social entities and the pattern of inter-relationships among them. What the relationship means varies, from those of social nature, such as kinship or friendship among people, to that of transactional nature, such as trading relationship between countries. Despite the variability in semantics, social networks share a common structure in which social entities, generically termed *actors*, are inter-linked through units of relationship between a pair of actors known as: *tie*, *link*, or *pair*. By representing actors as nodes and ties as edges, social network can be represented as a graph.

A constructed social network can be analyzed for many useful insights. For instance, the important actors in the network, those with the most connections, or the greatest influence [10, 17], can be found. Alternatively, it may be the connection paths between actors that are of interest. Analysts may look for the shortest paths [25], or the most novel types of connections [13]. Sometimes, the focus may even be on finding subgroups, subsets of the network that are especially cohesive or interesting [3, 15].

Knowledge of social networks is useful in various application areas. In law enforcement concerning organized crimes such as drugs and money laundering [25] or terrorism [12], knowing how the perpetrators are connected to one another would assist the effort to disrupt a criminal act or to identify additional suspects. In commerce, viral marketing exploits the relationship between existing and potential customers to increase sales of products and services [10, 17]. Members of a social network may also take advantage of their connections to get to know others, for instance through web sites facilitating networking or dating among their users [5].

Despite its many uses, social network is difficult to construct if only because a tie between a pair of actors is a property of the pair, rather than inherent to either actor. Collecting data on n actors quickly degenerates into finding the properties of $\frac{n(n-1)}{2}$ pairs of actors. Furthermore, the classical means of collecting such data by social scientists, though done carefully and reliably, are painstaking and time-consuming, involving questionnaires, interviews, direct observations, manual sifting through archival records, or various experiments [23]. This is fine for research studies experimenting on a small, controlled group of actors. However, wide application of social network analysis requires the ability to construct a large social network quickly, which can be achieved through computational methods capable of dealing with a huge amount of data.

In this paper, we look at computationally mining

*School of Computer Engineering, Nanyang Technological University, Singapore.

†Centre for IT Services, Nanyang Technological University, Singapore

‡Institute for Infocomm Research, Singapore

social network from spatio-temporal data. Each unit of such data has an associated location and time. Assuming that each data unit can also be attributed to a specific individual, the subset of data for an individual describes the series of locations visited by the individual over time. For example, such data may be obtained by tracking physical locations of moving objects, or by logging cyber locations visited by Internet users. Taking it a step further, we propose using spatio-temporal co-occurrence as a basis for inferring association between people. It is intuitive to think that co-occurring items may be related in some way, just as thunder's always following lightning tells us that they are somehow related. In this context, spatio-temporal co-occurrence is roughly defined as occurring together in space and time. By taking into account the frequency and the intensity of co-occurrences among people as they move around, we believe some knowledge about their relationships can be mined from the data.

Before stating the problem in earnest, we first enumerate our assumptions on the characteristics of data that we are dealing with. For a database $\mathcal{D}$, each tuple $d \in \mathcal{D}$ has the form of $d = \langle a, t, s \rangle$, where $d.a$ identifies an actor uniquely and $d.s$ indicates the location of this actor at time $d.t$. Though in reality seamlessly continuous, time is expressed as discrete values at a particular granularity (e.g., seconds). Furthermore, it is assumed that each data unit may be generated anytime, rather than only at strictly regular intervals as found in time series. Meanwhile, we model space as a collection of semantic locations, which may be physical locations such as rooms and buildings or cyber locations such as web addresses and domains. It is more practical to assume a semantic rather than a more refined coordinate space, which would have been more difficult to record accurately and would have required a mapping to correlate a coordinate to an actual location. Small-scale efforts to track locations, such as within building complexes, would likely settle for semantic location as it would be both easier and more useful to know that a person is at a particular room than at a given xyz coordinate. A semantic location may be expressed at several levels of granularity (e.g., room or building, web address or domain), and would also have a natural meaning indicating the purpose of the location, which would render a co-occurrence there even more meaningful.

We describe the problem in general terms as follows:

Given: *spatio-temporal database $\mathcal{D}$ as described*

Find: *social network graph $G(G_V, G_E)$, where:*

G_V is the set of nodes/actors in G

G_E is the set of edges/links in G based on spatio-temporal co-occurrences among actors

The problem as previously stated further spawns two subproblems:

1. *How are links between actors defined based on spatio-temporal co-occurrences?*
2. *How can such links be efficiently found?*

The rest of the paper is organized as follows. In Section 2, we survey various criteria used in mining social networks, and further explore the idea of co-occurrence. With respect to the first subproblem, in Section 3 we define a particular co-occurrence termed *spatio-temporal event* and describe how it could be used to infer links between actors. As a solution to the second subproblem, an algorithmic approach to mine social network based on events is presented in Section 4. Subsequently, Section 5 describes a real-life spatio-temporal data collected from web usage logs, and presents the experimental results on that data. Finally, in Section 6, we summarize our findings and suggest some directions for future work.

2 Related Work

Before we embark on a discussion on various ways of constructing social network, we first run through terms commonly used in social network literatures. As earlier mentioned, a node in a social network graph is termed an *actor*. A *tie* relates two actors. Like edges of a graph, ties could be directed or undirected, and they could be dichotomous (present or absent) or valued (weighted). There may be many types of ties (e.g., kinship, friendship) and the collection of all ties of the same type is a *relation*. *Social network* is a finite set of actors and all the relations among them. If all actors in a network are of the same type, the network is a *one-mode network*. Otherwise, a network with n types of actors is an *n-mode network*. These terms will be used throughout the rest of this paper.

2.1 Mining Social Network In addition to co-occurrence, these three criteria have also been used to infer ties between actors: self-report, communication, and similarity.

Self-report uses only links reported by individual actors. Such links are directed and naturally subjective. There could be cases where a claim of a tie is not reciprocated to the same extent, if at all. Classical tools like questionnaires and interviews are based on this principle [23]. Homepages or profile pages in community-centric sites such as LiveJournal weblogs [11] or Friendster networking site [5] commonly display a self-professed list of friends within the community. A similar idea is also present in the buddy list feature of Instant Messaging systems [18].

Communication, defined generally as transfer of information or resources, is common among socially related people. Inversely, evidence of communication may indicate association. Among others, such evidence may come from computer-mediated communication. Examples where the electronic trails of communication can be traced include emails [21], newsgroups [3], and Instant Messaging [18]. Links based on communication are directed, from the originator to the recipient.

Similarity has its foundation on the sociological idea that friends tend to be alike [6]. This leads to the premise that the more people have in common, the likelier it is that they are related. For example, homepages with similar textual content and linkages may represent a group of related individuals [1]. Other forms of similarity include having the same communication partners [21] and sharing the same opinions or areas of interest [17]. Similarity-based links are undirected.

Co-occurrence assumes that if several entities occur together more frequently than random chance alone would allow, they may be associated. Like similarity, it is by nature undirected and symmetric. The work on connection subgraph [8] uses a huge network whose ties identify pairs of people whose names are frequently mentioned together on the same webpages. Co-authorship networks, in turn, relate people who co-author the same publications together [10, 13].

Note that the above criteria for mining social network are seldom applied on spatio-temporal data. Of the four, co-occurrence is the most akin to such data as its meaning carries the sense of being together, possibly in space and time. This motivates us to pursue further the idea of spatio-temporal co-occurrence as a basis for inferring association.

2.2 Mining Co-occurrences With regards to time and space, there are four different ways to define co-occurrence: basic, when neither time nor space is considered; temporal and spatial, when only time or space is considered respectively; and spatio-temporal when both time and space are considered together.

Basic co-occurrence is mined from a database of discrete instances within which a few items co-occur. A major body of work on this type of co-occurrence is association rule mining [2]. For a given set of n items, $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$, a transaction is a discrete instance, uniquely identified by a *pid*, within which a subset of these items, $p \subseteq \mathcal{I}$, co-occur. The database to be mined is the set of all transactions $\mathcal{P}$. A pattern of co-occurrence involves a subset of items, $q \subseteq \mathcal{I}$, called an itemset, whose support is a function of $|\{p \in \mathcal{P} \mid q \subseteq p\}|$. If the support of q is beyond a given threshold, it is a frequent itemset, which is to say that members of q are deemed to be associated with one another.

Temporal co-occurrence does not assume that the data already has clearly-defined transactions. Instead, every tuple $\langle t, i\rangle$ is an item i occurring at time t, and subsets are derived using the time component of tuples. In the simplest case, two tuples $\langle t_1, i_1\rangle$ and $\langle t_2, i_2\rangle$ support an itemset $\{i_1, i_2\}$ if $|t_1 - t_2| \leq \delta$, for a given interval bound δ. Sequential patterns [4] and frequent episodes [16] not only care about the interval bound, but also the order at which items occur within the bound. Inter-transaction rules [14] would also demand the distance between occurrences of those items. Most strictly of all, time series patterns [7] specify an ordered series of items/values at regular intervals of time.

Spatial co-occurrence is aptly termed co-location. Each tuple $\langle s, i\rangle$ indicates that item i occurs at location s. Given the variety of spatial models [19], the notion of being co-located depends on the specific definition of space, from adjacent nodes in a graph space to items enclosed within a distance radius in a Euclidean space, but commonly captures the sense of being close by or neighboring. Another variation arises from how to define transaction-like instances over space. One way is to specify a reference feature (e.g., a lake), and treat each instance of that feature (and its neighboring items) as a transaction [9]. Alternatively, the space can be discretized by using a sliding window [20]. Yet another way is to materialize transactions wherever neighboring items are found, but constrain the multiplicity of the same item in many transactions [20].

Spatio-temporal co-occurrence deals with tuples with both space and time components. Despite the variability of spatial and temporal co-occurrences leading to the guess that there will be many ways to define spatio-temporal co-occurrence, current works in the area mainly focus on the time series approach. Spatio-temporal data is treated as a collection of time series of each item's wherebeing over time. Using time series similarity measures such as Euclidean [24] or LCSS [22] distance functions, the distance between two time series is evaluated. If it is below a certain threshold, the time series are considered similar enough, and the corresponding items are deemed to be co-occurring.

So far we have been mentioning co-occurrences of items, rather than actors. This is because the idea of spatio-temporal co-occurrence as indicative of association of social nature has not been much explored. Group pattern mining [24] is the closest to this direction, arguing that people who are consistently moving together may belong to a group. However, its focus is less on constructing a network formed by pairwise ties than on finding groups of increasing cardinality. Moreover, it assumes data in the form of time series of coordi-

nate locations, which leads to different formulations of the problem, and correspondingly to different solutions. In the next section, we propose a problem formulation based on irregular timing and semantic location that attempts to find pairwise ties between actors on the basis of spatio-temporal co-occurrence.

3 Mining Social Network from Events

3.1 Basic Events Just as an instance of co-occurring items is given the special term transaction in association rules, an instance of co-occurring actors is termed an *event* in social network terms. The work on inferring an association between actors through their participation in events is grounded in the affiliation network [23]. An affiliation network is a two-mode network, with a set of actors and a set of events connected by actor-event links. An event is any social collectivity of several actors, including conferences, games, social events, or meetings. An actor's affiliation to an event, by registration or attendance, establishes an actor-event link between the actor and the event.

By its act of bringing actors together, an event serves as conduit for resource transfer, or simply as a basis for interaction to take place. For example, conferences gather academicians around the world to exchange knowledge and build contacts. Linkages established through events can be interpreted in two ways. Firstly, an event enhances pairwise interactions between its members, in which case an event with n members gives rise to $\frac{n(n-1)}{2}$ actor-actor links. The second interpretation treats each event as a simultaneous linkage between all of its n members, much like a hyperedge connecting n vertices. Taking the first interpretation, which is more synchronous with most works in social network, an actor-actor tie between a pair of actors is said to exist if there is at least one event that the two actors are both affiliated to. Moreover, the number of such events can be taken as the weight of the tie. The collection of all such ties make up the social network.

In Figure 1, we give an example of an affiliation network, represented as a bipartite graph involving four actors $\{a_1, a_2, a_3, a_4\}$ and three events $\{e_1, e_2, e_3\}$. We have actors a_1 and a_2 affiliated to events e_1 and e_2, a_3 to e_2 and e_3, and a_4 to e_3. Based on their common affiliation to events, the actors can be linked in a social network as shown in Figure 2. Each actor-actor link indicates that two actors participate in at least one event together, and the weight of each link refers to the number of such events. Only a_1 and a_2 are linked by two events (e_1 and e_2), while the other pairs have only one event each. These figures illustrate how a two-mode affiliation network between actors and events can be transformed into a one-mode network of actors.

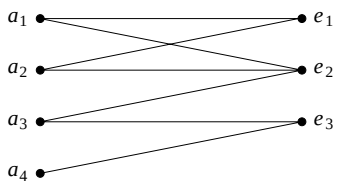

Figure 1: Two-Mode Affiliation Network

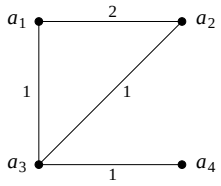

Figure 2: One-Mode Social Network

3.2 Spatio-Temporal Events Constructing such event-based networks as the above requires clearly-defined events gleaned from such sources as membership or attendance registers. Although spatio-temporal data as described does not carry information on events attended by actors, it can still tell us the spatio-temporal behavior of actors. We focus on one particular behavior: that actors may congregate together when engaged in social interactions. A corollary to that is that social events would produce spatio-temporal co-occurrences. Taking such co-occurrences as surrogates for events, we define a *spatio-temporal event* as a spatio-temporal co-occurrence that may have arisen from an underlying social interaction. Heretofore, we refer to nominal event as *basic event* and spatio-temporal event as just *event*.

Now we are ready to formally define an event. We adopt the notations for data as described before, where each tuple $d \in \mathcal{D}$ has the form of $\langle a, t, s \rangle$, identifying the location s of actor a at time t. Then, for a specified semantic location granularity and a time interval δ_{max}, an event is formally defined in the following way.

DEFINITION 3.1. *Event is a subset of tuples, $e \subseteq \mathcal{D}$, meeting the following conditions:*

- $\forall d_i, d_j \in e,\ d_i.s = d_j.s$,
  i.e., tuples are of the same location
- $\forall d_i, d_j \in e,\ |d_i.t - d_j.t| \leq \delta_{max}$,
  i.e., tuples are separated in time by at most δ_{max}
- $|\{d.a \mid d \in e\}| \geq 2$,
  i.e., tuples represent two or more actors
- *for any event $e' \subseteq \mathcal{D}$, $(e' \subseteq e) \vee (e \subseteq e') \Rightarrow (e = e')$,*
  i.e., each event is maximal

As required by the first two conditions, the semantic location granularity and time interval δ_{max} specify the constraints of a co-occurrence. Respectively, they indicate the furthest actors could be in space and time to still co-occur with each other. They could be as restrictive or as permissive as required to still render a co-occurrence meaningful in the sense of inducing some association among actors. Insisting on perfectly exact co-occurrences would be neither possible nor practical. Given the continuity of time and the limited sensitivity of even the finest measuring device, we cannot claim "at exactly the same time" with certainty anyway. While we use equality of locations to define co-occurrence, any location with non-zero area already implies a degree of tolerance. In any case, even if possible, exact co-occurrences might be rare. Furthermore, by allowing this tolerance to be varied, a suitable tolerance level can be chosen for the particular needs of the data.

The third condition requires that an event must concern more than one actor. It is obvious that for a spatio-temporal co-occurrence to be a surrogate for an actual interaction, it must involve at least two actors.

Finally, requiring each event to be maximal places a constraint on the multiplicity of tuples in being included in more than one event. Its purpose is to ensure, as much as possible, that each event stands for a single underlying interaction. Generally, events may overlap in terms of tuples, but they ought not to be subsets of another to avoid endless creation of events without gaining any additional information. The downside of this is that due to the constraint of δ_{max}, a long-running interaction may get split into several overlapping events.

Having defined event, we then enumerate some notations related to events. The set of all events defined over database $\mathcal{D}$ is denoted as $\mathcal{E}$. An event $e \in \mathcal{E}$ has several properties. The set of distinct actors represented by tuples in an event is its *actor set*, $e.\mathcal{A} = \{d.a \mid d \in e\}$, with size $|e.\mathcal{A}|$. An event's *start time*, $e.t^- = min_{d \in e}(d.t)$, and *end time*, $e.t^+ = max_{d \in e}(d.t)$, are the times of its earliest and latest tuples respectively. Correspondingly, its *duration*, $e.\delta = |e.t^- - e.t^+|$, is the distance between the two. The *area* $e.\triangle$ of an event measures the scope of its semantic location. We do not specify the exact form of this property, other than that for two locations, where one contains the other, the area value should be monotonic with respect to the granularity of the semantic location, i.e., the containing should have no smaller area than the contained. Lastly, its *weight* $e.w$ is a goodness measure related to the quality of relationship among actors of that event.

At this point we would like to note that perhaps with the exception of self-report, all other association criteria do not guarantee certainty in inferring association between actors. What they do is to mine evidence of association and assign a weight to each tie to indicate the likelihood of there being an actual association. Beyond a certain value where we feel confident enough about the existence of a tie, the weight may in turn assume the role of indicating the relationship strength of that tie. For affiliation network, every basic event is as good as any other as no effort is made to favor one over another. In our case, events possess some spatial and temporal information, which we will attempt to use to assign weights in ways that would boost the ability of events to both predict a relationship and measure the strength of that relationship. Towards this extent, we adopt the notions of *precision* and *uniqueness*.

Precision of an event refers to the quality of co-occurrence that defines the event with respect to tolerances in space and time. Intuitively, a co-occurrence at a finer granularity of space or time will also be valid at a coarser granularity. Besides being harder to achieve, the former is a more "exact", and thus a higher-quality, co-occurrence.

$$e.w_{p-s} = \frac{\frac{1}{e.\triangle}}{\max_{e' \in \mathcal{E}} (\frac{1}{e'.\triangle})} \tag{3.1}$$

Spatial precision of an event, denoted as $e.w_{p-s}$, measures how closely in space actors are from each other when participating in an event. This measure should be directly related to the granularity of the event's location, which in turn is related to the event's area $e.\triangle$. We define spatial precision as the inverse of an event's area, normalized with respect to the maximum such value among all events, as described by the equation Eq. 3.1. By this token, events held in smaller locations would be more precise than those in larger ones. The value of $e.w_{p-s}$ falls in the range of $(0, 1]$.

$$e.w_{p-t} = 1 - \frac{e.\delta}{(\delta_{max} + \delta_{unit})} \tag{3.2}$$

Temporal precision can similarly be based on duration $e.\delta$. Some may argue that very short durations are less important since they may have arisen from chance alone. That might have been valid if we know how long an actor stays at each location, which unfortunately we cannot know for certain given the assumption that the data is a set of snapshots, rather than a regular stream, of actors' locations. Instead, we take the reverse position that a shorter duration leads to a greater confidence that a co-occurrence has actually taken place. Besides, chance co-occurrences should be infrequent and can be removed accordingly. As such, temporal precision is defined as given in Eq. 3.2, giving a maximum value of 1 to events with perfect co-occurrence ($e.\delta = 0$). Addition of a unit of time δ_{unit} to the denominator is meant

to ensure a non-zero minimum value for cases where $e.\delta = \delta_{max}$. The value of δ_{unit} depends on the smallest division of time supported in the data, but in most cases we simply use $\delta_{unit} = 1$, assuming δ_{max} is expressed as a multiple of δ_{unit}. It follows that the range of $e.w_{p-t}$ falls in the range of $(0, 1]$.

Uniqueness is based on the idea that co-occurrence on a more unique premise is likely to indicate a stronger association. Unique items are deemed better because there is a lower probability of them being shared given their somewhat rarer occurrences. For instance, it has been suggested that commonly-shared features are weaker than unique features in predicting similarity-based association [1], or that novel, exclusive connections are more interesting than common ones [13].

$$e.w_{u-s} = 1 - \frac{|\{e' \in \mathcal{E},\ e' \neq e \mid e'.s = e.s\}|}{|\mathcal{E}|} \tag{3.3}$$

Spatial uniqueness refers to how unique the location of an event is among other events. Intuitively, if a location where not many other events take place is chosen, the interaction implied might also be of a different, and possibly more interesting nature. For an event e, its spatial uniqueness is given in Eq. 3.3. By counting only events other than itself, we ensure a non-zero minimum value such that $0 < e.w_{u-s} \leq 1$.

$$e.w_{u-t} = 1 - \frac{|\{e' \in \mathcal{E}, e' \neq e \mid e'.[t^-, t^+] \cap e.[t^-, t^+] \neq \emptyset\}|}{|\mathcal{E}|} \tag{3.4}$$

Temporal uniqueness, for all the same reasons, has an effect that is similar and parallel to spatial uniqueness. Instead of having a unique location, an event is temporally unique if it happens when relatively few other events are taking place. With a low level of background activity, it is an even lower probability that an event happens by coincidence. Furthermore, with such a judicious choice of time it is even likelier that the event is of a higher significance. However, in contrast to the semantic location case where overlap can be verified by equality, two events overlap temporally if they share at least a non-zero period of time. If the period of time covered by an event is denoted as an interval $e.[t^-, t^+]$, the function for temporal uniqueness is given in Eq. 3.4. As is the case with spatial uniqueness, we have $0 < e.w_{u-t} \leq 1$.

$$e.w = e.w_{p-s} \times e.w_{p-t} \times e.w_{u-s} \times e.w_{u-t} \tag{3.5}$$

Finally, we express an event's overall weight in Eq. 3.5 as the product of the above measures. Having non-zero value for each measure would prevent any one measure from nullifying the contribution of the other measures. Since each measure falls between 0 exclusive and 1 inclusive, the weight will also be in that range, $0 < e.w \leq 1$. Thus an event's weight can be interpreted as the probability that the event predicts an actual association between participating actors, or the strength of such a predicted association.

Dealing with semantic locations, we have defined spatial co-occurrence not in terms of distance interval, but in terms of a specified semantic location granularity. In reality, a database may have tuples with locations of varying granularity. For example, postal address has a home unit, city, state, and country. We may choose to restrict co-occurrence to the finest granularity only (e.g., home unit). However, what would be more practical is to allow co-occurrences to take place at various granularities, and to give events fair weights reflecting the weaker precision of a coarser granularity. Noting that locations of different granularities may subsume each other (e.g., home unit is contained in a city), we would not want to redundantly count events. In other words, two actors co-occurring in a city is redundant when we know they are in the same room. Towards this extent, we define a subevent-superevent relationship among events.

DEFINITION 3.2. *An event e_{sub} is a subevent of another event e_{sup}, or alternatively e_{sup} is a superevent of e_{sub}, if the following conditions are met:*

- $(e_{sup}.\triangle > e_{sub}.\triangle) \ \wedge \ (e_{sup}.s \text{ contains } e_{sub}.s)$
- $(e_{sup}.t^- \leq e_{sub}.t^-) \ \wedge \ (e_{sub}.t^+ \leq e_{sup}.t^+)$
- $e_{sub}.\mathcal{A} \subseteq e_{sup}.\mathcal{A}$

The first condition captures the sense that subevent-superevent relationship arises from differing location granularity. The latter two conditions are consequences of the first. By requiring co-occurrence at a finer granularity, a subevent is naturally more restrictive, and its duration and actor set are necessarily subsets of those of its superevent. Note that the relevance of these terms would come in later when we establish links based on events.

3.3 Event-based Links With some variation, we can derive a social network between pairs of actors based on spatio-temporal events in much the same way as that based on basic events. In affiliation network, a basic event is known for certain to be either present or absent. On the other hand, spatio-temporal events are inferred from the data and assigned a weight in the range of 0 to 1. If we take this weight as the probability that an event predicts an association, we may want to accept only events whose weight is above a certain threshold as capable of supporting links between actors.

DEFINITION 3.3. *An event e supports a link $\langle a_x, a_y \rangle$ between two actors, a_x and a_y, if $(\{a_x, a_y\} \subseteq e.\mathcal{A}) \wedge (e.w \geq min\_event\_weight)$, for a given threshold min_event_weight.*

For any given pair, there may well be more than one such event. We can then group together all such events as the *event set* of the pair. Furthermore, owing to the multi-granularity of semantic locations, we should take care to only include the most specific subevents supporting a linkage between the pair.

DEFINITION 3.4. *For a link $\langle a_x, a_y \rangle$, its event set is $\mathcal{E}_{\langle a_x, a_y \rangle} \subseteq \mathcal{E}$, such that:*

- $\mathcal{E}_{\langle a_x, a_y \rangle} = \{e \in \mathcal{E} \mid (\{a_x, a_y\} \subseteq e.\mathcal{A}) \wedge (e.w \geq min\_event\_weight)\}$
- $\forall e \in \mathcal{E}_{\langle a_x, a_y \rangle} \nexists e' \in \mathcal{E}_{\langle a_x, a_y \rangle}$, e' *is a subevent of* e

Greater cardinality of an event set means that more events support the association between the corresponding pair. Consequently, not only the link between the pair is more likely, it is also likely to be stronger. In order to factor this in the relationship strength of a pair, we define a link weight for a pair of actors $\langle a_x, a_y \rangle$ as the summation of the weight of the events in its event set, as given in Eq. 3.6. With that, we can then decide whether or not a link between a pair of actors exists.

$$\langle a_x, a_y \rangle.w = \sum_{e \in \mathcal{E}_{\langle a_x, a_y \rangle}} e.w \tag{3.6}$$

DEFINITION 3.5. *A link $\langle a_x, a_y \rangle$ between two actors, a_x and a_y, exists if $\langle a_x, a_y \rangle.w \geq min\_link\_weight$, for a given threshold min_link_weight.*

Keeping in mind that a social network is composed of links between pairs of actors, we restate the problem definition given previously as follows:

Given: *database $\mathcal{D}$, maximum duration δ_{max}, and thresholds min_event_weight, min_link_weight*

Find: *social network graph $G(G_V, G_E)$, where:*

$$G_E = \{\langle a_x, a_y \rangle \mid \langle a_x, a_y \rangle.w \geq min\_link\_weight\}$$
$$G_V = \{a \mid \exists \langle a_x, a_y \rangle \in G_E,\ a \in \{a_x, a_y\}\}$$

4 Algorithmic Approach

Since the database involved could be huge, in terms of the number of tuples and actors, the social network construction problem as posed in the above requires computational means to solve. Our proposed algorithm runs in two major phases. In the first phase, events are constructed from the database, and in the second phase, links are derived from those events.

ALGORITHM 4.1. Construction of Events

Input: database $\mathcal{D}$, time interval δ_{max}

Output: events $\mathcal{E}$

```
1: E = ∅,  E_cand = ∅,
2: for each tuple d ∈ D in the order of d.t do
3:    for each event e ∈ E_cand, (d.t − e.t⁻ > δ_max) do
4:       if (|e.A| > 1) ∧ (∄e' ∈ E, (e ⊆ e')) then
5:          E = E ∪ {e}
6:       end if
7:       E_cand = E_cand − {e}
8:    end for
9:    if ∄e ∈ E_cand, (e.s = d.s) ∧ (e.t⁻ = d.t) then
10:      create new event e = {d}
11:      E_cand = E_cand ∪ {e}
12:   end if
13:   for each event e ∈ E_cand, (e.s = d.s) do
14:      e = e ∪ {d}
15:   end for
16: end for
17: return E
```

The algorithm for the first phase is presented in Algorithm 4.1. The objective of this phase is to scan the database $\mathcal{D}$ and construct the set of events $\mathcal{E}$. Tuples of the database $\mathcal{D}$ are traversed in the chronological order. Recently created events that may still be affected by incoming tuples are first temporarily stored in $\mathcal{E}_{cand}$. This temporary store continually discards events whose temporal properties do not allow them to accept more tuples, i.e., when an event's duration would breach the limit of δ_{max}. Events with more than one actor and that are not just subsets of existing events in $\mathcal{E}$ are transferred into $\mathcal{E}$. A new event is created when a new location or a new timestamp is seen. Recent events in the temporary store $\mathcal{E}_{cand}$ of the same location as the incoming tuple are updated. Finally, the set of events $\mathcal{E}$ is returned as output of this phase. The time complexity of this phase is $O(|\mathcal{D}|)$, determined mainly by the outermost loop as the inner loops all concern $\mathcal{E}_{cand}$ whose cardinality is constrained by the value of δ_{max}.

Events created in the first phase are fed into the next phase, where the weights of these events are evaluated. The algorithm for the second phase is given in Algorithm 4.2. The first outermost loop iterates through the set of events $\mathcal{E}$. Each of the four measures, followed by the overall weight, of each event is computed. If an event's weight is beyond the threshold min_event_weight, it is eligible to support links among pairwise actors in its actor set. Each such pair is inserted into the set of candidate links $G_{E_{cand}}$. The algorithm also keeps the event set of each

pair updated and ensures that only the most specific subevents are used. At the end of the first outermost loop, we have the set of candidate links G_{Ecand} and the event sets of these candidate links. Subsequently, in the second outermost loop, the algorithm traverses through G_{Ecand}, first evaluating the weight of each candidate link and then verifying whether the weight is beyond the threshold *min_link_weight*. Such links are inserted into G_E, and the corresponding actors are inserted into G_V. As the computation of an event's weight may require traversal through $\mathcal{E}$, for instance to determine uniqueness the number of other events sharing similar spatial or temporal properties needs to be counted, the time complexity of the first outermost loop is $O(|\mathcal{E}|^2)$. The second outermost loop is clearly $O(|G_{Ecand}|)$. Hence this phase's time complexity is $O(|\mathcal{E}|^2 + |G_{Ecand}|)$.

ALGORITHM 4.2. Construction of Links

Input: events $\mathcal{E}$, *min_event_weight*, *min_link_weight*

Output: actors G_V, links G_E

1: $G_V = \emptyset,\ G_E = \emptyset,\ G_{Ecand} = \emptyset,$
2: **for** each event $e \in \mathcal{E}$ **do**
3: compute $e.w_{p-s}$, $e.w_{p-t}$, $e.w_{u-s}$, $e.w_{u-t}$
4: $e.w = e.w_{p-s} \times e.w_{p-t} \times e.w_{u-s} \times e.w_{u-t}$
5: **if** $e.w \geq min\_event\_weight$ **then**
6: **for** each pair $a_x, a_y \in e.\mathcal{A}$ **do**
7: $G_{Ecand} = G_{Ecand} \cup \{\langle a_x, a_y \rangle\}$
8: **if** $\nexists e' \in \mathcal{E}_{\langle a_x, a_y \rangle}, (e'$ subevent of $e)$ **then**
9: remove superevents of e from $\mathcal{E}_{\langle a_x, a_y \rangle}$
10: $\mathcal{E}_{\langle a_x, a_y \rangle} = \mathcal{E}_{\langle a_x, a_y \rangle} \cup \{e\}$
11: **end if**
12: **end for**
13: **end if**
14: **end for**
15: **for** each link $\langle a_x, a_y \rangle \in G_{Ecand}$ **do**
16: $\langle a_x, a_y \rangle.w = \sum_{e \in \mathcal{E}_{\langle a_x, a_y \rangle}} (e.w)$
17: **if** $\langle a_x, a_y \rangle.w \geq min\_link\_weight$ **then**
18: $G_E = G_E \cup \{\langle a_x, a_y \rangle\}$
19: $G_V = G_V \cup \{a_x, a_y\}$
20: **end if**
21: **end for**
22: return G_V, G_E

Combining the two phases is as easy as executing them in series. Initiated with database $\mathcal{D}$ as well as input parameters δ_{max}, *min_event_weight*, and *min_link_weight*, the combined algorithm outputs G_V and G_E, respectively the sets of nodes and edges of the desired social network graph G, at an overall time complexity of $O(|\mathcal{D}| + |\mathcal{E}|^2 + |G_{Ecand}|)$.

5 Experimental Results

5.1 Experimental Data For the experiments, we use a real-life data on webpages requested by wireless computer users at our university campus. The data is collected from firewall server logs over the whole month of August 2004. Each tuple contains a timestamp, a user login name, and a URL address. In total, there are about 4 million tuples, 2656 users, and 1.3 million URL addresses out of 58 thousand distinct URL domains. This data complies with the characteristics of spatio-temporal data that we expect. Actors are identified by their login names, which are anonymized to protect privacy. A tuple is generated whenever a URL request is made, and is timestamped up to the second ($\delta_{unit} = 1s$). URL addresses can be modeled as semantic locations and their directory structure corresponds to the multi-granularity of such locations. Here, we focus only on the URL domain level, and all the addresses are stripped down to their domains.

Though different from geographical locations, URL domains still have inherent semantic meaning in both the words that make up the domains as well as in the pages or sites that they represent. We figure that this semantic meaning would still render co-occurrences at such locations as potentially indicative of association between users. People do interact in the Internet and people visiting similar pages may have similar interests, may be collaborating on a task, may be influencing each other by recommending Internet resources, etc. All these carry a sense of association between people, the very thing we would like to mine.

5.2 Varying Parameters Through several experiments, we vary the input parameters to the algorithm to see the behavior or the properties of events and links that are generated. At any one time, we vary one parameter while fixing the rest. When fixed, the parameters would have the following values. We choose the maximum duration δ_{max} to be 2 hours which we deem a reasonable value for a meaningful co-occurrence at a URL domain. Expressing it in terms of δ_{unit}, we have $\delta_{max} = 7200s$. Next, we assume that all events should matter, so *min_event_weight* $= 0$. Meanwhile, we do not specify the value of *min_link_weight*, and first look only at candidate links, which are basically pairs of actors participating in at least one event together.

At first, we vary the size of the data along the chronological axis, while fixing the other parameters as mentioned in the above. Starting with a single day, we incrementally increase the input data, each time by adding a day's worth of data. Figure 3 shows the effect of increasing the number of tuples $|\mathcal{D}|$ to the number of events $|\mathcal{E}|$ and candidate links $|G_{Ecand}|$ generated.

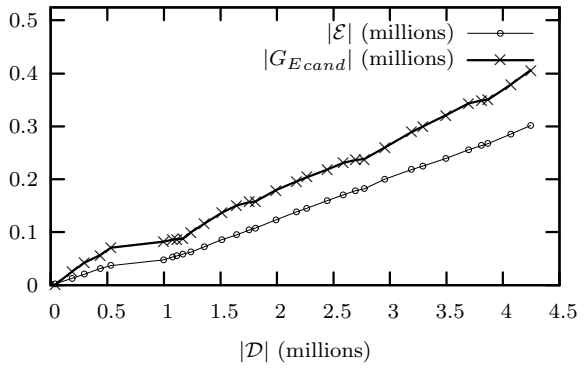

Figure 3: $|\mathcal{E}|$, $|G_{Ecand}|$ vs. $|\mathcal{D}|$

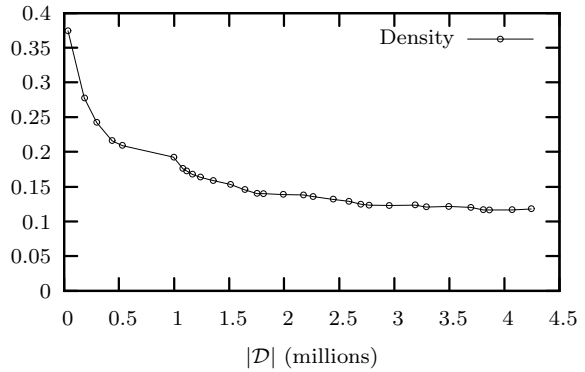

Figure 4: Density vs. $|\mathcal{D}|$

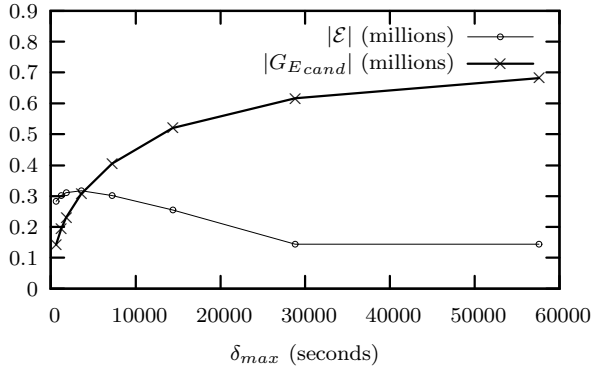

Figure 5: $|\mathcal{E}|$, $|G_{Ecand}|$ vs. δ_{max}

The latter two would have an effect on the algorithm's second phase's complexity. Clearly, there is an almost linear relationship between the data size varied along time and the number of events, which is actually quite intuitive as over an extended period of time, the rate at which events are taking place in real life should be relatively constant. Meanwhile, the steady increase in the number of candidate links is due to a different reason. Since not all actors are active all the time, extending the period of time being covered increases the likelihood that we catch their occurrences when they happen to be active. However, we expect that in the long run this number would level off as over a long enough period of time every actor would have had at least one event with every one of their acquaintances.

The increasing number of candidate links in Figure 3 is due to the increasing number of actors becoming active as data size increases. To take into account increases in the number of actors, we look at the density of the graph formed by the candidate links. Given n actors, the maximum possible number of links would be $\frac{n(n-1)}{2}$. Density of a graph is the fraction of the number of existent links over the maximum possible number [23]. For the graph formed from candidate links, the density value would in fact be $\frac{2\times|G_{Ecand}|}{n(n-1)}$. In Figure 4, we track this density as we increase the data size, which indirectly also increases the number of actors. We see that as the data size increases, the density at first decreases and then slowly converges to around 0.1, implying that for a large data size only about one-tenth of all possible links would be candidate links. This shows that the number of candidate links is related to the number of actors, and once the number of actors converges, so should the number of candidate links.

Next, we use the full data size, and again fix $min\_event\_weight = 0$ for the same reasons as the above. As δ_{max} is varied from 10 minutes to 16 hours, initially there is a growth in the number of events materialized, as shown in Figure 5. This is because a larger δ_{max} is more permissive that even tuples separated relatively widely in time can still form an event. Beyond a certain value of δ_{max}, the number of events begin to decline before leveling off as a very large δ_{max} results in several events of the same location being combined with one another to form a long-running event. In contrast, the number of candidate links continues to increase, though at a decreasing rate and eventually leveling off. Larger values of δ_{max} tend to be less restrictive in creating events, leading to more pairs having at least one common event.

Previously, no min_link_weight has been specified and we have only looked at candidate links. If specified, candidate links whose weight exceeds this

min_link_weight	$\|G_E\|$
0	406078
1	71866
5	5299
10	1569
20	421
30	176
40	85
50	44
60	25
70	13
80	7
90	3
100	2

Figure 6: *min_link_weight* vs. $|G_E|$

min_link_weight value would be included as links in the social network. Using the full data size, and parameter values $\delta_{max} = 7200s$ and *min_event_weight* $= 0$, we vary *min_link_weight* from 0 to 100 to get the number of links produced at each threshold value. Although we expect that the number of links ($|G_E|$) will be lower at higher threshold values, Figure 6 further shows that the drop in the number of links caused by increasingly higher thresholds is extremely precipitous. With 2656 actors, there could be up to $(\frac{1}{2})(2656)(2655)$ or 3.5 million links. Less than 12% of that number is supported by any event at all (*min_link_weight* $= 0$). By *min_link_weight* $= 20$, the number of links has dropped to hundreds. We recall that in affiliation network, a link is weighted by the number of basic events, and is deemed to exist if there is at least one basic event supporting that link. In our case, a link's weight is the sum of its supporting events' weights, with each event having a weight between 0 and 1. Setting *min_link_weight* $= 1$ would be equivalent to requiring at least one basic event to establish a link. In turn each *min_link_weight* value can be interpreted as the number of full events required to instill enough confidence that a pair of actors are actually related. There is a direct tradeoff between the confidence in links and the number of links that can be included in the social network graph.

Notably, with rare occurrences of links with very high weight while the vast majority of links have very low weight (0 to 1), the distribution of link weights seems to approximate the Zipfian [26] distribution, a distribution that has been shown by many other social networks as well [17].

Common Features	*Random-Pairs*	*Event-Pairs*
at least 0	100%	100%
at least 1	49%	90%
at least 2	12%	23%
at least 3	1%	3%

Figure 7: Demographic Similarity

5.3 Demographic Similarity Ideally, the links generated by the proposed event-based method can be verified to a high degree of confidence by gathering feedback from the concerned actors directly. Unfortunately, that has not been feasible in our case as there are strict restrictions on approaching the actors included in the data directly to protect their privacy. However, we have a limited demographic information about the actors. Relying on the idea that related actors tend to be similar (Section 2.1), we wish to check whether the event-based links that we have generated would show greater demographic similarity than links drawn at random.

The demographic features that can be obtained for each actor include her *major* (e.g., business, computer science), *status* (e.g., undergraduate, postgraduate, staff), and *year of entry* into the university. For each link between a pair of actors, we count the number of feature values the two actors have in common (from 0 to 3). For comparison, we draw two sets of links. *Random-Pairs* consists of 100 links formed by drawing a pair of actors at random from the pool of actors. *Event-Pairs* consists of 100 links with the highest link weights among the links generated by the proposed method run on the full data with parameters $\delta_{max} = 7200s$ and *min_event_weight* $= 0$. For each set, we count the number of links having at least 0 to 3 feature values in common. The results shown in Figure 7 confirm that at high threshold values, there tends to be a greater amount of demographic similarity in the event-based links than in the random links. While not spectacular by itself, *Event-Pairs* shows a not insignificant increase over *Random-Pairs*. On average, *Event-Pairs*' similarity percentages are about twice those of *Random-Pairs*.

To illustrate highly similar event-based pairs, in Figure 8 we use as examples the three pairs from the *Event-Pairs* set that have all three demographic features in common. Demography refers to the status, major, and year of entry of both actors in a pair. The first pair of actors, referred to as $\langle a_1, a_2 \rangle$, are both MBA students beginning in 2004. Events involving these actors include, but not exclusively, the given URL domains. The first two domains, those of Yahoo! India and Rediff.com (an India-based portal), indicate their Indian origin. The next two domains tell us their

Pairs	*Demography*	*Sample URL Domains*
$\langle a_1, a_2 \rangle$	Postgraduate (MBA) Business 2004	login.india.yahoo.com www.rediff.com www.carinfousa.com cdn.movies-etc.com
$\langle a_3, a_4 \rangle$	Postgraduate (Research) Biology 2003	www.ecallchina.com www.sohu.net nar.oupjournals.org www.ncbi.nlm.nih.gov
$\langle a_5, a_6 \rangle$	Postgraduate (Research) Civil Engin. 2003	eae.seu.edu.cn www.sciencedirect.com www.sina.com.cn xintv.xinhuanet.com

Figure 8: Highly Similar Event-based Pairs

common interests in car prices in the USA and in online movies. The second pair of actors, $\langle a_3, a_4 \rangle$, are both research students in biology beginning in 2003. The first two sample domains are China-based portals, again revealing their country of origin. Those are followed by domains belonging to the Nucleic Acid Research Journal and National Center for Biotechnology Information respectively, which suggest their similar research areas. The last pair of actors, $\langle a_5, a_6 \rangle$, are research students in civil engineering beginning in 2003. Both actors might have affiliation to South East University in China, as indicated by the first domain listed. Both have also used ScienceDirect, an online library portal, presumably for their research. Finally, the next two domains are again those of popular China-based portals. In these cases, we are fairly confident that actors in each pair are likely to know each other given such similar areas of interests, countries of origin, and demographic features. Furthermore, they also show that the event-based approach seems to be able to generate results that correlate with those from the similarity-based approach.

Rather than claiming the results above as absolute, we caution that the demographic information used to derive similarities is rather limited and that similarity on its own is not an authoritative method for verification. Nevertheless, we are still encouraged that the correlation between our proposed co-occurrence-based method with another, similarity-based method seems to indicate that our approach has a promising research potential.

6 Conclusion

In this paper, we introduce the problem of mining social network from spatio-temporal data. We propose using spatio-temporal co-occurrence as a basis for inferring associations of social nature. This is facilitated by our novel definition of spatio-temporal events, which we then use to derive event-based links between pairs of actors. After providing an algorithm that mines the desired event-based social network in two phases, we present our experiments on a real-life data on web usage logs collected at our own university. Comparison of the links produced by our proposed method and another, similarity-based method shows an encouraging result, especially keeping in mind that it has a real potential of generating large social networks from spatio-temporal data quickly for industrial or commercial uses.

There are many avenues for future works. Our current approach could be fine-tuned by investigating other factors that may help boost the quality of events and by learning from the results on different datasets. Faster algorithms that can deal with much larger data size or data streams would increase the utility of the proposed approach. The constructed social network can also be analyzed for useful patterns or insights such as temporal evolution or periodicity of relationships. Finally, we would also look at how patterns of mobility in spatio-temporal data, concerning speeds and sequence of locations traversed, may be used in mining social networks.

Acknowledgments

We would like to thank the Centre for IT Services, Nanyang Technological University, for providing us with the experimental data used in this paper, as well as the Agency for Science, Technology and Research (A*STAR) for partially funding the work presented in this paper through an A*STAR Graduate Scholarship.

References

[1] L. A. Adamic and E. Adar, *Friends and neighbors on the web*, Social Networks, 25 (2003), pp. 211–230.

[2] R. Agrawal and R. Srikant, *Fast algorithm for mining association rules*, VLDB, (1994), pp. 487–499.

[3] R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu, *Mining newsgroups using networks arising from social behavior*, WWW, (2003), pp. 688–703.

[4] R. Agrawal and R. Srikant, *Mining sequential patterns*, ICDE, (1995), pp. 3–14.

[5] D. M. Boyd, *Friendster and publicly articulated social networking*, in Conf. on Human Factors and Computing Systems, (2004), pp. 1279–1282.

[6] K. Carley, *A theory of group stability*, American Sociological Review, 56 (1991), pp. 331-354.

[7] G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth, *Rule discovery from time series*, KDD, (1998), pp. 27–31.

[8] C. Faloutsos, K. S. McCurley, and A. Tomkins, *Connection subgraphs in social networks*, in Workshop on

Link Analysis, Counterterrorism, and Privacy (in conj. with SDM), (2004).

[9] K. Koperski and J. Han, *Discovery of spatial association rules in geographic information databases*, SSD, (1995), pp. 47—66.

[10] D. Kempe, J. Kleinberg, and E. Tardos, *Maximizing the spread of influence through a social network*, KDD, (2003), pp. 137–146.

[11] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, *Structure and evolution of blogspace*, CACM, 47 (2004), pp. 35–39.

[12] V. E. Krebs, *Mapping networks of terrorist cells*, Connections, 24 (2002), pp. 43–52.

[13] S. Lin and H. Chalupsky, *Unsupervised link discovery in multi-relational data via rarity analysis*, ICDM, (2003), pp. 171–178.

[14] H. Lu, L. Feng, and J. Han, *Beyond intratransaction association analysis: mining multidimensional intertransaction association rules*, ACM TOIS, 18 (2000), pp. 423–454.

[15] M. Mukherjee and L. B. Holder, *Graph-based data mining on social networks*, in Workshop on Link Analysis and Group Detection (in conj. with KDD), (2004).

[16] H. Mannila, H. Toivonen, and A. I. Verkamo, *Discovering frequent episodes in sequences*, KDD, (1995), pp. 210–215.

[17] M. Richardson and P. Domingo, *Mining knowledge-sharing sites for viral marketing*, KDD, (2002), pp. 61–70.

[18] J. Resig, S. Dawara, C. M. Homan, and A. Teredesai, *Extracting social networks from instant messaging populations*, in Workshop on Link Analysis and Group Detection (in conj. with KDD), (2004).

[19] S. Shekhar, S. Chawla, S. Ravada, A. Fetterer, X. Liu, and C. Lu, *Spatial databases - accomplishments and research needs*, IEEE TKDE, 11 (1999), pp. 45–55.

[20] S. Shekhar and Y. Huang, *Discovering spatial co-location patterns: a summary of results*, SSTD, (2001), pp. 236–256.

[21] M. F. Schwartz and D. C. M. Wood, *Discovering shared interests using graph analysis*, CACM, 36 (1993), pp. 78–89.

[22] M. Vlachos, G. Kollios, and D. Gunopulos, *Discovering similar multidimensional trajectories*, ICDE, (2002), pp. 673-684.

[23] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.

[24] Y. Wang, E. Lim, and S. Hwang, *On mining group patterns of mobile users*, DEXA, (2003), pp. 287–296.

[25] J. Xu and H. Chen, *Fighting organized crimes: using shortest-path algorithms to identify associations in criminal networks*, Decision Support Systems, 38 (2004), pp. 473–487.

[26] G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Boston, MA, 1949.