Finding Your Friends and Following Them to Where You Are

Adam Sadilek
Dept. of Computer Science
University of Rochester
Rochester, NY, USA
sadilek@cs.rochester.edu

Henry Kautz
Dept. of Computer Science
University of Rochester
Rochester, NY, USA
kautz@cs.rochester.edu

Jeffrey P. Bigham
Dept. of Computer Science
University of Rochester
Rochester, NY, USA
jbigham@cs.rochester.edu

ABSTRACT

Location plays an essential role in our lives, bridging our online and offline worlds. This paper explores the interplay between people's location, interactions, and their social ties within a large real-world dataset. We present and evaluate Flap, a system that solves two intimately related tasks: link and location prediction in online social networks. For link prediction, Flap infers social ties by considering patterns in friendship formation, the content of people's messages, and user location. We show that while each component is a weak predictor of friendship alone, combining them results in a strong model, accurately identifying the majority of friendships. For location prediction, Flap implements a scalable probabilistic model of human mobility, where we treat users with known GPS positions as noisy sensors of the location of their friends. We explore supervised and unsupervised learning scenarios, and focus on the efficiency of both learning and inference. We evaluate Flap on a large sample of highly active users from two distinct geographical areas and show that it (1) reconstructs the entire friendship graph with high accuracy even when no edges are given; and (2) infers people's fine-grained location, even when they keep their data private and we can only access the location of their friends. Our models significantly outperform current comparable approaches to either task.

Categories and Subject Descriptors

H.1.m [**Information Systems**]: Miscellaneous

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Location modeling, link prediction, social networks, machine learning, graphical models, visualization, Twitter

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.

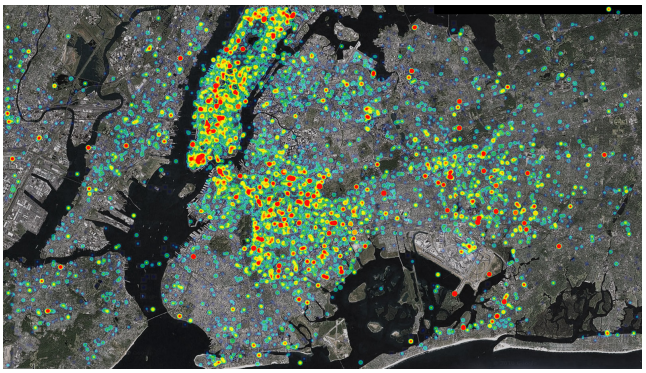

Figure 1: A snapshot of a heatmap animation of Twitter users' movement within New York City that captures a typical distribution of geo-tagged messaging on a weekday afternoon. The hotter (more red) an area is, the more people have recently tweeted from that location. Full animation is at http://cs.rochester.edu/u/sadilek/research/.

1. INTRODUCTION

Our society is founded on the interplay of human relationships and interactions. Since every person is tightly embedded in our social structure, the vast majority of human behavior can be fully understood only in the context of the actions of others. Thus, not surprisingly, more and more evidence shows that when we want to model the behavior of a person, the best predictors are often not based on the person herself, but rather on her friends, relatives, and other *connected* people. For instance, behavioral patterns of people taking taxis, rating movies, choosing cell phone providers, or sharing music are often best predicted by the habits of related people, rather than by the attributes of the individual such as age, ethnicity, or education [3, 24].

Until recently, it was nearly impossible to gather large amounts of data about the connections that play such important roles in our lives. However, this is changing with the explosive increase in the use, popularity, and significance of *online* social media and *mobile* devices.[1] The online aspect makes it practical to collect vast amounts of data, and the mobile element bridges the gap between our online and offline activities. Unlike other computers, phones are aware of the location of their users, and this information is often included in users' posts. In fact, major online social networks

[1] http://www.comscore.com/

are fostering location sharing. Twitter added an explicit GPS tag that can be specified for each tweet (AKA Twitter message update) in early 2010 and is continually improving the location-awareness of its service. Google+, Facebook, FourSquare, and Gowalla allow people to share their location, and to "check-in" at venues. With Google Latitude and Bliin, users can *continually* broadcast their location.

Thus, we now have access to colossal amounts of real-world data containing not just the text and images people post, but also their location. Of course, these three data modalities are not necessarily mutually independent. For instance, photos are often GPS-tagged and locations can also be mentioned, or alluded to, in text.

While the information about users' *location* and *relationships* is important to accurately model their behavior and improve their experience, it is not always available. This paper explores novel techniques of inferring this latent information from a stream of message updates. We present a unified view on the interplay between people's location, message updates, and their social ties on a large real-world dataset. Our approaches are robust and achieve significantly higher accuracy than the best currently known methods, even in difficult experimental settings spanning diverse geographical areas.

1.1 Significance of Results

Consider the task of determining the exact geographic location of an arbitrary user of an online social network. If she routinely geo-tags her posts and makes them public, the problem is relatively easy. However, suppose the location information is hidden, and you only have access to public posts by her friends. By leveraging social ties, our probabilistic location model—the first component of this work—infers where any given user is with high accuracy and fine granularity in both space and time *even when the user keeps his or her posts private*. Since this work shows that once we have location information for some proportion of people, we can infer the location of their friends, one can imagine doing this recursively until the entire target population is covered. To our knowledge, no other work attempts to predict locations in a comparably difficult setting.

The main power of our link prediction approach—the second major component of this work—is that it accurately reconstructs the entire friendship graph even when no "seed" ties are provided. Previous work either obtained very good predictions at the expense of large computational costs (*e.g.*, [28]), thereby limiting those approaches to very small domains, or sacrificed orders of magnitude in accuracy for tractability (*e.g.*, [19, 7]). By contrast, we show that our model's performance is comparable to the most powerful relational methods applied in previous work [28], while at the same time being applicable to large real-world domains with tens of millions (as opposed to hundreds) of possible friendships. Since our model leverages users' locations, it not only encompasses virtual friendships, but also begins to tie them together with their real-life groundings.

Prediction of people's location and social ties—especially when considered together—has a number of important applications. They range from improved local content with better social context, through increased security (both personal and electronic) via detection of abnormal behavior tied with one's location, to better organization of one's relationships and connecting virtual friendships with the real-world. We note that even when friends participate in the same social networking platform, their relationship may not be exposed—either because the connections are hidden or because they have not yet connected online. Flap can also help contain disease outbreaks [12]. Our model allows identification of highly mobile individuals as well as their most likely meeting points, both in the past and in the future. These people can be subsequently selected for targeted treatment or preemptive vaccination. Given people's inferred locations, and limited resource budget, a decision-theoretic approach can be used to select optimal emergency policy. Clearly, strong privacy concerns are tied to such applications, as we discuss in the conclusions.

2. RELATED WORK

Recent research in **location-based reasoning** explored harnessing data collected on regular smart phones for modeling human behavior [10]. Specifically, they model individuals' general location from nearby cell towers and Bluetooth devices at various times of day. Eagle *et al.* show that predicting if a person is at home, at work, or someplace else can be achieved with more than 90% accuracy. Besides scalability and practicality—social network data is much more readily available than cell phone logs—our work differs in that we include dynamic relational features (location of friends), focus on a finer time granularity, and consider a substantially larger set of locations (hundreds per user, rather than three). Additionally, the observations in our framework (people's self-published locations) are significantly noisier and less regular than cell tower and Bluetooth readings. Finally, our location estimation applies even in situations, where the target people decide to keep their data private.

Backstrom *et al.* predict the home address of Facebook users based on provided addresses of one's friends [2]. An empirically extracted relationship between geographical distance and the probability of friendship between pairs of users is leveraged in order to find a maximum likelihood assignment of addresses to hidden users. The authors show that their method of localizing users is in general more accurate than an IP address-based alternative. However, even their strongest approach captures only a single static home location for each user and the spatial resolution is low. For example, less than 50% of the users are localized within 10 miles of their actual home. By contrast, we consider much finer temporal resolution (20 minute intervals) and achieve significantly greater spatial precision, where up to 84% of people's exact dynamic location is correctly inferred.

Very recently, Cho *et al.* focus on modeling user location in social networks as a dynamic Gaussian mixture, a generative approach postulating that each check-in is induced from the vicinity of either a person's home, work, or is a result of social influence of one's friends [6]. By contrast, our location model is inherently discrete, which allows us to predict the *exact* location rather than a sample from a generally high-variance continuous distribution; operates at a finer time granularity; and learns the candidate locations from noisy data. Furthermore, our approach leverages the complex temporal and social dependencies between people's locations in a more general, discrete fashion. We show that our model outperforms that of Cho *et al.* in the experiments presented below.

A number of geolocating applications demonstrate emerg-

ing privacy issues in this area. The most impactful ones are arguably Creepy[2], ICanStalkU.com, and PleaseRobMe.com (currently disabled). The purpose of these efforts is to raise awareness about the lack of location privacy in online social networks. Given a username from an online social network, Creepy aggregates location information about the target individual from her GPS-tagged posts and photos, and displays it on a map. ICanStalkU.com scans public Twitter timeline, and extracts GPS metadata from uploaded photos, which often reveal people's current location without them realizing it. PleaseRobMe.com used to extract people's geographic check-ins that imply they are not at home and therefore vulnerable to burglaries.

However, all these applications work only with *publicly available* data. By contrast, this paper shows that we can *infer* people's precise location even when they keep their data private, as long as some of their friends post their location publicly. Therefore, simply turning off the geolocation features of your phone—which may seem to be a reliable way to keep your whereabouts secret—does not really protect your privacy unless your friends turn theirs off as well.

While our work concentrates on Twitter users, recent research shows that the predictability of human mobility remains virtually unchanged across a number of demographical attributes such as age and gender [27]. This strongly suggests that our approach achieves similar accuracy for other geographical areas and different samples of users. Finally, we note that although it has been shown that possibly as many as 34% of accounts provide either wrong or misleading symbolic (*e.g.*, city, state) location information in their profiles, our work is largely shielded from this phenomenon since we focus only on raw GPS data that can be more readily verified and is not fed into a geocoder [15].

The problem of **link prediction** has been studied in a large number of domains and contexts; here we mention the ones that are most relevant to our work. Liben-Nowell *et al.* models the *evolution* of a graph solely from its topological properties [19]. The authors evaluate a number of methods of quantifying the probability of a link in large graphs, and conclude that while no single technique is substantially better than the rest, their models outperform a random predictor by a significant margin. This shows that there is important information to be mined from the graph structure alone.

An alternative approach is to leverage the topology as well as attributes of the individual entities [28]. They model friendships of students in an online university community using relational Markov networks. Similarly to our approach, their probabilistic model encompasses a number of features, some of which are based on the attributes of individual users while others model the structure of the friendship graph. Their inference method is standard belief propagation (BP), whereas we develop an efficient and specialized version of BP, which in practice quickly converges. Their domain contains only several hundred candidate social ties. This size restriction is apparently due to the computational challenges posed by their relational model. We, in contrast, consider thousands of individuals who can be connected in arbitrary fashion, which results in tens of millions potential friendships. Furthermore, Taskar *et al.* assume that some friendships are given to the model at testing time. In this work, we show that it is possible to achieve good performance even with no observed links.

Crandall *et al.* explore the relationship between co-location of Flickr users and their social ties [7]. They model the relationship as an exponential probability distribution and show it fits well to the observed, empirical distribution. They show that the number of distinct places where two users are co-located within various periods of time has the potential to predict a small fraction of the ties quite well. However, the recall is dramatically low. In their Flickr data, only 0.1% of the friendships meet the condition for being predicted with at least 60% confidence. By contrast, with our approach we can predict over 90% of the friendships with confidence beyond 80% (see Figure 4). This is consistent with our experiments, where we show that location alone is generally a poor predictor of friendship (consider the "commuter train" example described below on one end of the spectrum, and a pair of friends that never share their location on the other end). We therefore leverage textual similarity and network structure as well, and evaluate the predictive power of our model in terms of AUC while inferring the friendship graph. Additionally, our model does not require setting subtle parameters, such as cell size and time granularity. When we apply the method of Crandall *et al.* to our Twitter data, the results are (as one would expect) poor; see Figure 5 and its analysis in text.

The relationship between social ties and distance has recently received considerable attention [20, 2, 25]. Even though online interactions are in principle not hampered by physical distance, all works agree that in any social network studied, the probability of friendship generally decreases as the distance between people increases. However, a significant portion of social ties often cannot be explained by location alone. We observe the same pattern in our Twitter data and show that location can be augmented with text and structural features to infer social ties with high accuracy.

Backstrom *et al.* present a method for predicting links cast as a random walk problem [1]. A key difference between our approaches is that we can construct the entire social network with high accuracy even when *none* of the edges are observed, whereas Backstrom *et al.*'s approach depends upon already knowing most of the links in the network along with a set of source and candidate nodes, and only needs to predict relatively few new links. Furthermore, unlike our work, Backstrom *et al.*'s approach requires many parameters to be selected. In contrast with random walks, approaches related to our belief propagation method for enforcing and chaining soft transitive constraints have been validated in many areas in the machine learning literature, and are implicitly used in many works on link prediction as a way to solve the underlying probabilistic models [26, 28].

We note that no work to date focused on capturing both directions of the relationship between location and social ties. This paper concentrates on predicting, in turn, both location and social structure from the remaining data modality.

3. BACKGROUND

Our experiments are based on data obtained from Twitter, a popular micro-blogging service where people post at most 140 characters long message updates. The forced brevity encourages frequent mobile updates, as we show below. Relationships between users on Twitter are not necessarily sym-

[2]https://github.com/ilektrojohn/creepy

metric. One can follow (subscribe to receive messages from) a user without being followed back. When users do reciprocate following, we say they are *friends* on Twitter. There is anecdotal evidence that Twitter friendships have a substantial overlap with offline friendships [14]. Twitter launched in 2006 and has been experiencing an explosive growth since then. As of March 2011, approximately 200 million accounts are registered on Twitter.[3] For an excellent general overview of computational analysis of social networks at large see [11].

Decision trees are models of data encoded as rules induced from examples [4]. Intuitively, in the Twitter domain, a decision tree represents a series of questions that need to be asked and answered in order to estimate the probability of friendship between any two people, based on their attributes. During decision tree learning, features are evaluated in terms of information gain with respect to the labels and the best candidates are subsequently selected for each inner node of the tree. Our implementation uses *regression* decision trees, where each leaf contains the probability of a friendship. As described below, we also employ decision trees for feature selection, since they intrinsically rank features by their information content.

Belief propagation (BP) is a family of message passing algorithms that perform inference in graphical models. BP is proven to be exact and to converge for certain classes of graphs, such as trees, but its behavior on general cyclic graphs is poorly understood [23]. However, in many practical applications, BP performs surprisingly well [22].

Dynamic Bayesian networks (DBNs) are generative probabilistic graphical models of sequential data [21]. Nodes in the graph represent random variables and edges represent conditional dependencies. In a typical setting, a subset of the random variables is *observed*, while the others are *hidden* and their values have to be inferred. A DBN is composed of *slices*—in our case each slice represents a time interval. In order to specify a DBN, we either write down or learn intra- and inter-slice conditional probability distributions (CPDs). The intra-slice CPDs typically constitute the observation model while the inter-slice CPDs model transitions between hidden states.

There are a number of parameter learning and inference techniques for DBNs. In a supervised learning scenario, where the hidden labels are known at training time, maximum likelihood estimates can be calculated directly. On the other hand, when the state of the hidden nodes is not known, the CPDs have to be learned without supervision. We achieve this via expectation-maximization described below. Exact inference is usually intractable in general DBNs and one has to resort to sampling techniques such as Markov chain Monte Carlo. However, our model is sufficiently efficient to afford exact inference using dynamic programming.

In this work, we apply DBNs because they naturally model time series data (time flows in one direction), we can highly optimize both learning and inference. Since the hidden nodes in our models are discrete, we perform both parameter learning and exact inference efficiently by customized versions of the Baum-Welch algorithm and Viterbi decoding, respectively. For a detailed treatment of these methods see [17]. We exaplain how we apply DBNs to our Twitter domain in Section 5.2.

[3]http://www.bbc.co.uk/news/business-12889048

4. THE DATA

Using the Twitter Search API[4], we collected a sample of public tweets that originated from two distinct geographic areas: Los Angeles (LA) and New York City (NYC). The collection period was one month long and started on May 19 2010. Using a Python script, we periodically queried Twitter with requests of all recent tweets within 150 kilometers of LA's city center, and 100 kilometers within the NYC city center. In order to avoid exceeding Twitter's query rate limits and subsequently missing some tweets, we distributed the work over a number of machines with different IP addresses that asynchronously queried the server and merged their results. Twitter does not provide any guarantees as to what sample of existing tweets can be retrieved through their API, but a comparison to official Twitter statistics shows that our method recorded nearly all of the publicly available tweets in those two regions. Altogether, we have logged over 26 million tweets authored by more than 1.2 million unique users (see Table 1). To put these statistics in context, the entire NYC and LA metropolitan areas have an estimated population of 19 and 13 million people, respectively.[5]

In this work, we concentrate on accounts that posted more than 100 GPS-tagged tweets during the one-month data collection period. We refer to them as *geo-active users*.

New York City & Los Angeles Dataset

Unique users	1,229,611
Unique geo-active users	11,380
Tweets total	26,118,084
GPS-tagged tweets	7,566,569
GPS-tagged tweets by geo-active users	4,016,286
Unique locations	89,077
Significant locations	25,830
"Follows" relationships between geo-active users	123,182
"Friends" relationships between geo-active users	52,307

Table 1: Summary statistics of the data collected from NYC and LA. Geo-active users are ones who geo-tag their tweets relatively frequently (more than 100 times per month). Note that following reciprocity is about 42%, which is consistent with previous findings [18, 16]. Unique locations are the result of iterative clustering that merges (on a per-user basis) all locations within 100 meters of each other. Significant location is defined as one that was visited at least five times by at least one person.

5. THE SYSTEM: FLAP

Flap (Friendship + Location Analysis and Prediction), has three main components responsible for downloading Twitter data, visualization, and learning and inference. The data collection component was described in the previous section. Figure 2 shows Flap's visualization a sample of geo-active users in NYC. People are represented by pins on the map and the red links denote friendships (either ground truth or inferred). Beyond standard Google Maps user interface elements, the visualization is controlled via the black toolbar

[4]http://search.twitter.com/api/

[5]http://www.census.gov/popest/metro/

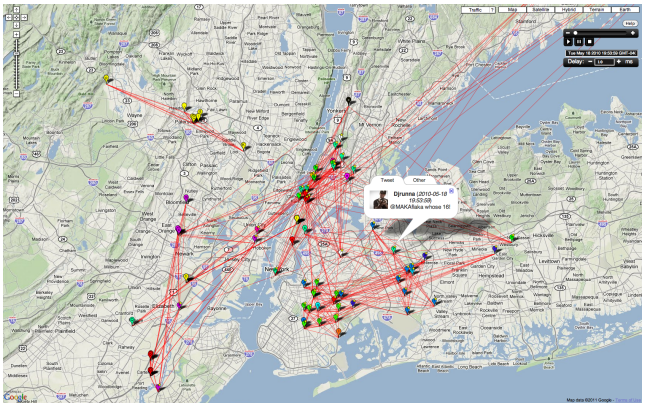

Figure 2: Flaps's visualization of a sample of geo-active friends in NYC. Red links between users represent friendships.

in the upper-right corner. Flap can animate arbitrary segments of the data at various speeds. Selecting a user displays additional information such as his profile, time and text of his recent tweets, and a more detailed map of his current surroundings.

Now we turn to the third—machine learning—module of Flap that has two main tasks. First, it is responsible for learning a model of people's friendships and subsequently revealing hidden friendships. And second, it learns models of users' mobility and predicts their location at any given time. We will now discuss these two tasks and our solutions in turn.

5.1 Friendship Prediction

The goal of friendship prediction is to reconstruct the entire social graph, where vertices represent users and edges model friendships. We achieve this via an iterative method that operates over the current graph structure and features of pairs of vertices. We first describe the features used by our model of social ties, and then focus on its structure, learning, and inference. In agreement with prior work, we found that no single property of a pair of individuals is a good indicator of the existence or absence of friendship [20, 6]. Therefore, we combine multiple disparate features—based on text, location, and the topology of the underlying friendship graph.

5.1.1 Features

The text similarity coefficient quantifies the amount of overlap in the vocabularies of users u and v, and is given by

$$\mathcal{T}(u,v) = \sum_{w \in W(u) \cap W(v) \setminus S} f_u(w) f_v(w), \tag{1}$$

where $W(u)$ is the set of words that appear in user u's tweets, S is the set of stop-words (it includes the standard stop words augmented with words commonly used on Twitter, such as RT, im, and lol), and $f_u(w)$ is the frequency of word w in u's vocabulary.

Interestingly, in the Twitter domain, the mentions tags (@) give a clue to user's friendships. However, in the experiments presented here, we *eliminate* all user names that appear in the tweets in order to report results that generalize to other social networks.

Our co-location feature ($\mathcal{C}$) is based on the observation that at least some people who are online friends also meet in the physical world [14]. We make an assumption that once a user tweets from a location, he or she remains at that location until they tweet again. Even though people generally do not tweet from every single place they visit, this approximate co-location measure still captures how much time pairs of users tend to spend close to each other. The co-location score is given by

$$\mathcal{C}(u,v) = \sum_{\ell_u, \ell_v \in L} \frac{t(\ell_u, \ell_v)}{d(\ell_u, \ell_v)}, \tag{2}$$

where L is the union of all locations from which users u and v send messages, $t(\ell_u, \ell_v)$ is the amount of time u spends at location ℓ_u while v is at location ℓ_v. In short, we add up the time overlaps two users spend at their respective locations and we scale each overlap by the distance between the locations. Thus, two individuals spending a lot of common time at nearby places receive a large co-location score, while people who always tweet from two opposite ends of a city have a small co-location score. We have implemented an efficient algorithm that calculates $\mathcal{C}(u,v)$ for a pair of users in time $O(n)$ where n is the minimum number of GPS-tagged messages created by either user u or v. Note that unlike previous work (*e.g.*, [7, 1]), our co-location feature is continuous and does not require discretization, thresholding, or parameter selection.

As a graph structure feature, we use the meet/min coefficient ($\mathcal{M}$) and its generalized version ($\mathcal{M}_{\mathbb{E}}$) defined in equations 3 and 4 respectively.

$$\mathcal{M}(u,v) = \frac{\big|N(u) \cap N(v)\big|}{\min\Big(\big|N(u)\big|, \big|N(v)\big|\Big)} \tag{3}$$

$$\mathcal{M}_{\mathbb{E}}(u,v) = \frac{\sum\limits_{n \in N(u) \cap N(v)} p_{nu} p_{nv}}{\min\Big(\sum\limits_{n \in N(u)} p_{nu}, \sum\limits_{n \in N(v)} p_{nv}\Big)} \tag{4}$$

$N(u)$ is the set of neighbors of node u and p_{nu} is the probability of edge (n,u). The standard meet/min coefficient counts the number of common neighbors of u and v (this quantity is equal to the number of triads that the edge (u,v) would complete, an important measure in structural balance theory [11]), and scales by the size of the neighborhood of either u or v, whichever is smaller. Intuitively, $\mathcal{M}(u,v)$ expresses how extensive is the overlap between friendlists of users u and v with respect to the size of the shorter friendlist. The expectation of the meet/min coefficient $\mathcal{M}_{\mathbb{E}}$ calculates the same quantities but in terms of their expected values on a graph where each edge is weighted by its probability. Neither measure depends on the existence or probability of edge (u,v) itself.

Since the $\mathcal{T}$ and $\mathcal{C}$ scores are always observed, we use a regression decision tree to unify them, in a pre-processing step, into one feature $\mathcal{DT}(u,v)$, which is the decision tree's prediction given $\mathcal{T}(u,v)$ and $\mathcal{C}(u,v)$. Thus, we end up with one feature function for the observed variables ($\mathcal{DT}$) and one for the hidden variables ($\mathcal{M}_{\mathbb{E}}$).

We have experimented with other features, including the Jaccard coefficient, preferential attachment, hypergeometric coefficient, and others. However, our work is motivated by having an efficient and scalable model. A decision tree-based feature selection showed that our three measures ($\mathcal{T}$, $\mathcal{C}$, and

$\mathcal{M}_{\mathbb{E}}$) jointly represent the largest information value. Finally, while calculating the features for all pairs of n users is an $O(n^2)$ operation, it can be significantly sped up via locality-sensitive hashing [8].

5.1.2 Learning and Inference

Our probabilistic model of the friendship network is a Markov random field that has a hidden node for each possible friendship. Since the friendship relationship is symmetric and irreflexive, our model contains $n(n-1)/2$ hidden nodes, where n is the number of users. Each hidden node is connected to an observed node ($\mathcal{DT}$) and to all other hidden nodes.

Ultimately, we are interested in the probability of existence of an edge (friendship) given the current graph structure and the pairwise features of the vertices (users) the edge is incident on. Applying Bayes' theorem while assuming mutual independence of features $\mathcal{DT}$ and $\mathcal{M}_{\mathbb{E}}$, we can write

$$\begin{aligned} &P(E=1|DT=d, M_{\mathbb{E}}=m) = \\ &= P(DT=d|E=1)P(M_{\mathbb{E}}=m|E=1)P(E=1)/Z \\ &= P(DT=d|E=1)P(E=1|M_{\mathbb{E}}=m)/Z \end{aligned} \tag{5}$$

where

$$Z = \sum_{i\in\{0,1\}} P(DT=d|E=i)P(E=i|M_{\mathbb{E}}=m).$$

E, DT, and $M_{\mathbb{E}}$ are random variables that represent edge existence, $\mathcal{DT}$ score, and $\mathcal{M}_{\mathbb{E}}$ score, respectively. In equation 5, we applied the equality

$$P(M_{\mathbb{E}}|E) = P(E|M_{\mathbb{E}})P(E)/P(M_{\mathbb{E}})$$

and subsequent simplifications so that we do not need to explicitly model $P(E)$.

At learning time, we first train a regression decision tree $\mathcal{DT}$ and prune it using ten-fold cross-validation to prevent overfitting. We also perform maximum likelihood learning of the parameters $P(DT|E)$ and $P(E|M_{\mathbb{E}})$. We chose the decision tree pre-processing step for several reasons. First, the text and location-based features considered individually or independently have very poor predictive power. Therefore, models such as logistic regression tend to have low accuracy. Furthermore, the relationships between the observed attributes of a pair of users and the their friendship is often quite complex. For example, it is not simply the case that a friendship is more and more likely to exist as people spend larger and larger amounts of time near each other. Consider two strangers that happen to take the same train to work, and tweet every time it goes through a station. Our dataset contains a number of instances of this sort. During the train ride, their co-location could not be higher and yet they are not friends on Twitter. This largely precludes success of classifiers that are looking for a simple decision surface.

At inference time, we use $\mathcal{DT}$ to make preliminary predictions on the test data. Next, we execute a customized loopy belief propagation algorithm that is initialized with the probabilities estimated by $\mathcal{DT}$ (see Algorithm 1). Step 6 is where an edge receives belief updates from the other edges as well as the $\mathcal{DT}$ prior. Even though the graphical model is dense, our algorithm converges within several hundred iterations, due in part to the sufficiently accurate initialization and regularization provided by the decision tree. Note that the algorithm can also function in an online fashion: as new active users appear in the Twitter public timeline, they are processed by the decision tree and added to Q. This is an attractive mode, where the model is always up to date and takes advantage of all available data.

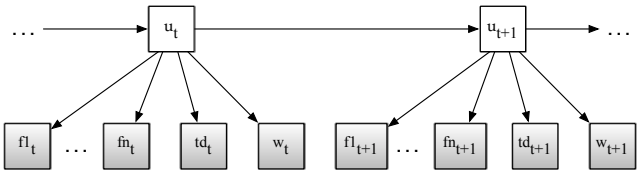

Figure 3: Two consecutive time slices of our dynamic Bayesian network for modeling motion patterns of Twitter users from n friends. All nodes are discrete, shaded nodes represent observed random variables, unfilled denote hidden variables.

Algorithm 1 : refineEdgeProbabilities(Q)

Input: Q: list containing all potential edges between pairs of vertices along with their preliminary probabilities
Output: Q: input list Q with refined probabilities

1: **while** Q has not converged **do**
2: sort Q high to low by estimated edge probability
3: **for each** $\langle e, P(e)\rangle$ **in** Q **do**
4: $dt \Leftarrow \mathcal{DT}(e)$
5: $m \Leftarrow \mathcal{M}_{\mathbb{E}}(e)$
6: $P(e) \Leftarrow \frac{P(DT=dt|E=1)P(E=1|\mathcal{M}_{\mathbb{E}}=m)}{\sum_{i\in\{0,1\}} P(DT=dt|E=i)P(E=i|\mathcal{M}_{\mathbb{E}}=m)}$
7: **end for**
8: **end while**
9: **return** Q

5.2 Location Prediction

The goal of Flap's location prediction component is to infer the most likely location of person u at any time. The input consists of a sequence of locations visited by u's friends (and for supervised learning, locations of u himself over the training period), along with corresponding time information. The model outputs the most likely sequence of locations u visited over a given time period.

We model user location in a dynamic Bayesian network shown in Figure 3. In each time slice, we have one hidden node and a number of observed nodes, all of which are discrete. The hidden node represents the location of the target user (u). The node td represents the time of day and w determines if a given day is a work day or a free day (weekend or a national holiday). Each of the remaining observed nodes ($f1$ through fn) represents the location of one of the target user's friends. Since the average node degree of geo-active users is 9.2, we concentrate on $n \in \{0, 1, 2, \ldots, 9\}$, although our approach works for arbitrary nonnegative values of n. Each node is indexed by time slice.

The domains of the random variables are generated from the Twitter dataset in the following way. First, for each user, we extract a set of distinct locations they tweet from. Then, we iteratively merge (cluster) all locations that are within 100 meters of each other in order to account for GPS sensor noise, which is especially severe in areas with tall buildings, such as Manhattan. The location merging is done separately for each user and we call the resulting locations

unique. We subsequently remove all merged locations that the user visited fewer than five times and assign a unique label to each remaining place. These labels are the domains of u and fi's. We call such places *significant*.

The above place indexing yields a total of 89,077 unique locations, out of which 25,830 were visited at least five times by at least one user. There were 2,467,149 tweets total posted from the significant locations in the 4 week model evaluation period. Table 1 lists summary statistics.

We model each person's location in 20 minute increments, since more than 90% of the users tweet with lower frequency. Therefore, the domain of the time of day random variable t_d is $\{0, \ldots, 71\}$ (total of 24/0.3 time intervals in any given day).

5.2.1 Learning

We explore both supervised and unsupervised learning of user mobility. In the earlier case, for each user, we train a DBN on the first three weeks of data with known hidden location values. In the latter case, the hidden labels are unknown to the system.

During *supervised* learning, we find a set of parameters (discrete probability distributions) θ that maximize the log-likelihood of the training data. This is achieved by optimizing the following objective function.

$$\theta^\star = \underset{\theta}{\operatorname{argmax}} \ \log\big(\Pr(x_{1:t}, y_{1:t}|\theta)\big), \tag{6}$$

where $x_{1:t}$ and $y_{1:t}$ represent the sequence of observed and hidden values, respectively, between times 1 and t, and $\theta^\star$ is the set of optimal model parameters. In our implementation, we represent probabilities and likelihoods with their log-counterparts to avoid arithmetic underflow.

For *unsupervised* learning, we perform expectation-maximization (EM) [9]. In the E step, the values of the hidden nodes are inferred using the current DBN parameter values (initialized randomly). In the subsequent M step, the inferred values of the hidden nodes are in turn used to update the parameters. This process is repeated until convergence, at which point the EM algorithm outputs a maximum likelihood point estimate of the DBN parameters. The corresponding optimization problem can be written as

$$\theta^\star = \underset{\theta}{\operatorname{argmax}} \ \log \sum_{y_{1:t}} \Pr(x_{1:t}, y_{1:t}|\theta)\big), \tag{7}$$

where we sum over all possible values of hidden nodes $y_{1:t}$. Since equation 7 is computationally intractable for sizable domains, we simplify by optimizing its lower bound instead, similar to [13].

The random initialization of the EM procedure has a profound influence on the final set of learned parameter values. As a result, EM is prone to getting "stuck" in a local optimum. To mitigate this problem, we perform deterministic simulated annealing [29]. The basic idea is to reduce the undesirable influence of the initial random set of parameters by "smoothing" the objective function so that it hopefully has fewer local optima. Mathematically, this is written as

$$\theta^\star(\tau_1, \ldots, \tau_m) = \underset{\theta}{\operatorname{argmax}} \ \tau_i \log \sum_{y_{1:t}} \Pr(x_{1:t}, y_{1:t}|\theta)^{\frac{1}{\tau_i}}. \tag{8}$$

Here, $\tau_1, \ldots, \tau_m$ is a sequence of parameters, each of which corresponds to a different amount of smoothing of the original objective function (shown in equation 7). The sequence is often called a *temperature schedule* in the simulated annealing literature, because equation 8 has analogs to free energy in physics. Therefore, we start with a relatively high temperature τ_1 and gradually lower it until $\tau_m = 1$, which recovers the original objective function.

5.2.2 Inference

At inference time, we are interested in the most likely explanation of the observed data. That is, given a sequence of locations visited by one's friends, along with the corresponding time and day type, our model outputs the most likely sequence of locations one visited over the given time period.

Flap runs a variant of Viterbi decoding to efficiently calculate the most likely state of the hidden nodes. In our model, Viterbi decoding is given by

$$y^\star_{1:t} = \underset{y_{1:t}}{\operatorname{argmax}} \ \log\big(\Pr(y_{1:t}|x_{1:t})\big), \tag{9}$$

where $\Pr(y_{1:t}|x_{1:t})$ is conditional probability of a sequence of hidden states $y_{1:t}$ given a concrete sequence of observations $x_{1:t}$ between times 1 and t.

In each time slice, we coalesce all observed nodes with their hidden parent node, and since we have one hidden node in each time slice, we apply dynamic programming and achieve polynomial runtimes in a way similar to [17]. Specifically, the time complexity of our inference is $O(T\,|Y|^2)$, where T is the number of time slices and Y is the set of possible hidden state values (potential locations).

Therefore, the overall time complexity of learning and inference for any given target user is $O(kT\,|Y|^2)$, where k is the number of EM iterations ($k = 1$ for supervised learning). This renders our model tractable even for very large domains that evolve over long periods of time with fine granularity. Next, we turn to our experiments, and analysis of results.

6. EVALUATION

For clarity, we discuss experimental results for each of the two Flap's tasks separately.

6.1 Friendship Prediction

We evaluate Flap on friendship prediction using two-fold cross-validation in which we train on LA and test on NY data, and vice versa. We average the results over the two runs. We varied the amount of randomly selected edges provided to the model at testing time from 0 to 50%.

Flap reconstructs the friendship graph well over a wide range of conditions—even when given no edges (Figure 4 and Table 2). It far outperforms the baseline model (decision tree) and the precision/recall breakeven points are comparable to those of [28], even though our domain is orders of magnitude larger and our model is more tractable.

We also compare our model to that of Crandall *et al.* [7], summarized in Section 2. Figure 5 shows the results of their contemporaneous events counting procedure on our Twitter data for various spatial and temporal resolutions. We see that in our dataset, the relationship between co-location and friendship is much more complex and non-monotonic as compared to their Flickr dataset. As a result, the predictive performance of Crandall *et al.*'s model on our data is poor. When probabilistically predicting social ties based on the number of contemporaneous events, the accuracy is 0.001%, precision 0.008, and recall 0.007 (in the best case,

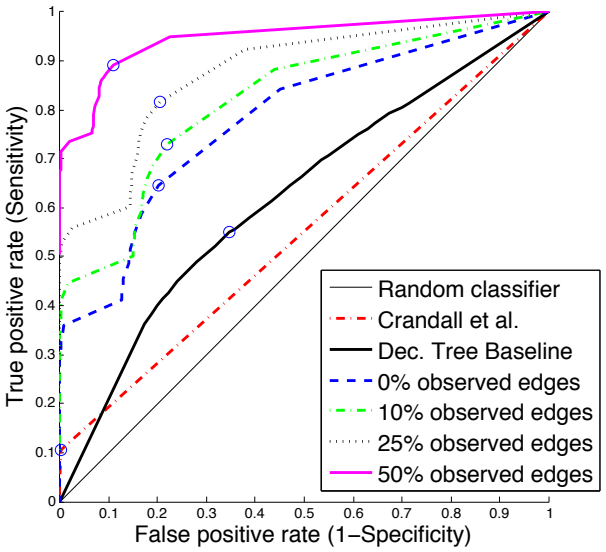

Figure 4: Averaged ROC curves for decision tree baseline, Crandall *et al.*'s model with the most favorable setting of parameters ($s = 0.001$ and $t = 4$ hours), and Flap.

	#E	0%	10%	25%	50%
AUC Flap	6.5×10^7	0.78	0.82	0.88	0.95
AUC Crandall *et al.*	6.5×10^7	0.55	-	-	-
P=R Flap	6.5×10^7	0.28	0.36	0.47	0.64
P=R Crandall *et al.*	6.5×10^7	0.05	-	-	-
P=R Taskar *et al.*	$\sim 4 \times 10^2$	N/A	0.47	0.58	0.73

Table 2: Summary of model evaluation. The #E column represents the number of candidate edges that exist in the social graph. The remaining columns denote the proportions of friendships given to the models at testing time. AUC is the area under the ROC curve; P=R denotes precision/recall breakeven points. All results are based on our Twitter dataset, except for the P=R results for Taskar *et al.*, which are based on their—much smaller—university dataset as their model does not scale to larger networks; see text for details.

where $s = 0.001$ and $t = 4$ hours). There are two conclusions based on this result. First, similarly to Liben-Nowell *et al.* [20], we observe that geographic distance alone is not sufficient to accurately model social ties. And second, looking at the performance of [7]'s approach on the Flick data comprising the entire world, versus its performance on our LA and NYC data, we see that inferring relationships from co-location data in dense and relatively small geographical areas can be a more challenging task. This is important, as the majority of population lives and interacts in such metropolitan areas. However, our work shows that when we leverage additional information channels beyond co-location, and embed them in a probabilistic model, we can infer social ties quite well.

In order to explore how our model performs in the context of *strong* ties, in both LA and NYC, we selected a subgraph that contains only active users who are members of a clique of size at least eight. We again evaluated via cross-validation as above. Flap reconstructs the friendship network of the

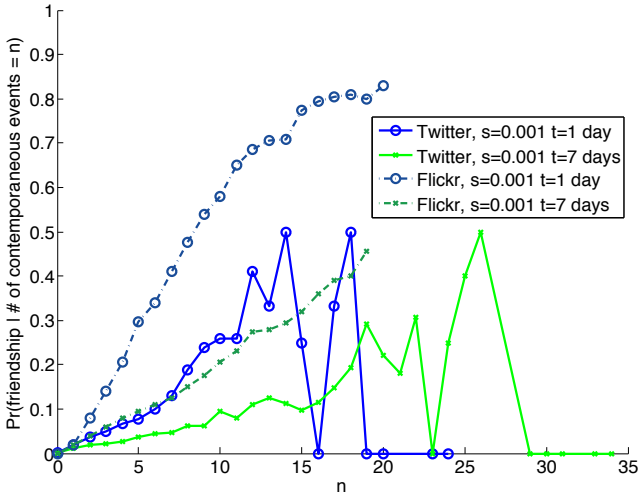

Figure 5: Comparison of the intensity of co-location of pairs of users versus the probability of their friendship in our Twitter and Crandall *et al.*'s Flickr datasets. We see that the relationship is more complex on Twitter, causing a simple model of social ties to achieve very low predictive accuracy. (s is the size of cells in degrees in which we count the co-located events and t is the time slack; compare with Figure 2 in [7].)

83 people with 0.92 precision and 0.85 recall, whereas the baseline decision tree achieves precision of 0.83 and recall of 0.51. Interestingly, the co-location feature plays a major role here because the cliques of friends spend most of their time in relatively small areas.

6.2 Location Prediction

Our evaluation is done in terms of *accuracy*—the percentage of timeslices for which the model infers the correct user location. We have a separate dynamic Bayesian network model for each user.

In order to evaluate the models learned in a *supervised* fashion, we train each model on three weeks worth of data (5/19/2010 00:00:00 – 6/8/2010 23:59:59) and test on the following fourth week (6/9/2010 00:00:00 – 6/15/2010 23:59:59). We always use the respective local time: PDT for LA, and EDT for NYC. We vary the number of friends (n) that we harness as sensors for each individual from 0 to 9. We always use the n most geo-active friends, and introduce a special missing value for users who have fewer than n friends. We evaluate the overall performance via cross-validation. In each fold of cross-validation, we designate a target user and run learning and inference for him. This process is repeated for all users, and we report the average results over all runs for a given value of n (Figure 6).

For models learned in an *unsupervised* manner, we also apply cross-validation as above. The hidden locations are learned via unsupervised clustering as described above. The temperature schedule for the EM procedure is given by $\tau_{i+1} = \tau_i \times 0.8$, with initial temperature $\tau_1 = 10$ (see equation 8). This results in calculating the likelihood at 11 different temperatures for each EM run. The EM procedure always converged within one thousand iterations, resulting in runtimes under a minute per user even in the largest domain.

We compare the results obtained by our DBN models to

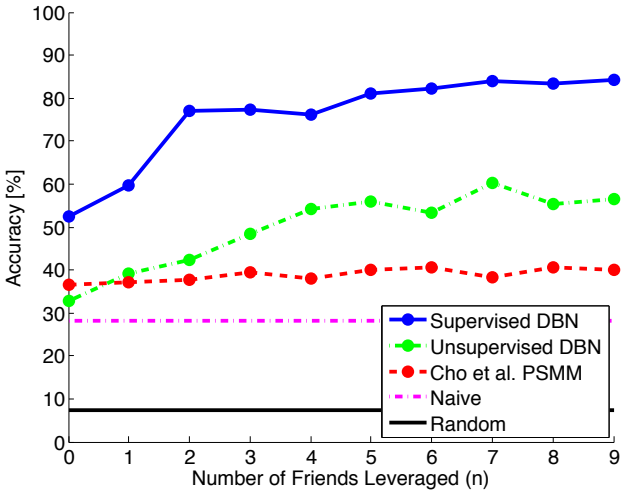

Figure 6: Predictive accuracy of location models. The performance of the two baseline models is by design independent of number of friends considered.

random and naïve baselines, and to the currently *strongest* mobility model of Cho *et al.* [6]. The random model is given the number of hidden locations for each user and guesses target user's location uniformly at random for each time slice. The naïve model always outputs the location at which the target user spends most of his time in the training data. We consider a prediction made by Cho *et al.*'s model accurate if it lies within 100 meters (roughly a city block) of the true user location.

Figure 6 summarizes the results. As expected, the supervised models perform better than their unsupervised counterparts. However, given the complexity of the domain and the computational efficiency of our models during training as well as testing, even the unsupervised models achieve respectable accuracy. The DBN approaches are significantly better than both random and naïve baselines, and they also dominate [6]'s social mobility model (PSMM) by a large margin. We believe this is mainly because people's mobility in condensed metropolitan areas often does not nicely decompose into "home" and "work" states, and the social influence on user location is not simply an attractive force (*i.e.*, one does not necessarily tend to appear closer to one's friends). For instance, consider two co-workers, one having a morning shift and the other a night shift in the same store. Their mobility is certainly intertwined, but the force between their location is a repulsive one, as when the first one is working, the other is sleeping at home. Unlike other approaches, our DBN model correctly learns such nonlinear patterns (both temporal and social).

The results are very encouraging. For example, even when the model is given information only about one's two friends, and no information about the target user (Unsupervised, $n = 2$ in Figure 6), it infers the correct location 47% of the time. As we increase the number of available friends to nine (Unsupervised, $n = 9$), we achieve 57% accuracy. When historical data about the mobility of the target user and his friends is available, we can estimate the correct location 77% of the time when two friends are available (Supervised, $n = 2$) and 84.3% with nine friends (Supervised, $n = 9$). As n increases, the accuracy generally improves. Specifically, we see that there is a significant boost from $n = 0$ to $n = 2$, after which the curves plateau. This suggests that a few active friends explain one's mobility well. We also see that simply outputting the most commonly visited location (Naïve) yields poor results since people tend to lead fairly dynamic lives.

7. CONCLUSIONS AND FUTURE WORK

Location information linked with the content of users' messages in online social networks is a rich information source that is now accessible to machines in massive volumes and at ever-increasing real-time streaming rates. This data became readily available only very recently. In this work, we show that there are significant patterns that characterize locations of individuals and their friends. These patterns can be leveraged in probabilistic models that infer people's locations as well as social ties with high accuracy. Moreover, the prediction accuracy degrades gracefully as we limit the amount of observed data available to the models, suggesting successful future deployment of Flap at a scale of an entire social network.

Our approach is quite powerful, as it allows us to reason even about the location of people who keep their messages and GPS data private, or have disabled the geo-features on their computers and phones altogether. Furthermore, unlike all existing approaches, our model of social ties reconstructs the entire friendship network with high accuracy even when the model is not "seeded" with a sample of known friendships. At the same time, we show that the predictions improve as we provide more observed edges at testing time.

By training the model on one geographical area and testing on the other using cross-validation (total of 4 million geo-tagged public tweets we collected from Los Angeles and New York City metropolitan areas), we show that Flap discovers robust patterns in the formation of friendships that transcend diverse and distant areas of the USA. We conclude that no single property of a pair of individuals is a good indicator of the existence or absence of friendship. And no single friend is a good predictor of one's location. Rather, we need to combine multiple disparate features—based on text, location, and the topology of the underlying friendship graph—in order to achieve good performance.

In our current work, we are extending the model to leverage the textual content of the tweets, as it contains hints about locations that are not captured by our existing features. We are currently exploring language understanding and toponym resolution techniques vital for tapping this information. We also focus on casting the two problems explored in this paper in a unified formalism and solving them *jointly*, perhaps in a recursive fashion.

We recognize that there are substantial ethical questions ahead, specifically concerning tradeoffs between the values our automated systems create versus user privacy. For example, our unsupervised experiments show that location can be inferred even for people who keep their tweets and location private, and thus may believe that they are "untrackable." These issues will need to be addressed in parallel with the development of our models. Other researchers have started exploring solutions to privacy concerns through data obfuscation [5].

However, we believe that the benefits of Flap—in helping to connect and localize users, and in building smarter systems—outweigh the possible dangers. There are many

exciting practical applications that have the potential to change people's lives that rely on location and link estimation. These include context-aware crowdsourcing, timely and accurate recommendations, better local content, disease prevention and containment, security, traffic modeling, emergency response, and others.

8. ACKNOWLEDGMENTS

This research was partly funded by ARO grant W911NF-08-1- 0242, ONR grant N00014-11-10417, OSD grant W81X WH-08-C0740, and a gift from the Kodak Company. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of any of these organizations.

9. REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[2] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, pages 61–70. ACM, 2010.

[3] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD*, pages 95–104, New York, NY, USA, 2007. ACM.

[4] L. Breiman et al. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

[5] A. B. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Ubicomp*, Ubicomp '10, pages 95–104, New York, NY, USA, 2010. ACM.

[6] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.

[7] D. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436, 2010.

[8] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[10] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[11] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[12] S. Eubank, H. Guclu, V. Anil Kumar, M. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429(6988):180–184, 2004.

[13] Z. Ghahramani. Learning dynamic Bayesian networks. In *Adaptive Processing of Sequences and Data Structures*, page 168. Springer, 1998.

[14] A. Gruzd, B. Wellman, and Y. Takhteyev. Imagining twitter as an imagined community. In *American Behavioral Scientist, Special issue on Imagined Communities*, 2011.

[15] L. Hong, B. Suh, and E. H. Chi. Tweets from justin bieber's heart: the dynamics of the "location" field in user profiles. In *ACM CHI*, 2011.

[16] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD/SNA-KDD*, pages 56–65, New York, NY, USA, 2007. ACM.

[17] M. Jordan. *Learning in graphical models*. Kluwer Academic Publishers, 1998.

[18] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *WWW*, April 2010.

[19] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58:1019–1031, May 2007.

[20] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11623, 2005.

[21] K. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[22] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *In Proceedings of Uncertainty in AI*, pages 467–475, 1999.

[23] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[24] A. S. Pentland. *Honest Signals: How They Shape Our World*. The MIT Press, 2008.

[25] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. *Proceedings of ICWSM*, 11, 2011.

[26] D. Smith and J. Eisner. Dependency parsing by belief propagation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 145–156. Association for Computational Linguistics, 2008.

[27] C. Song, Z. Qu, N. Blumm, and A. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018, 2010.

[28] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *in Neural Information Processing Systems*, 2003.

[29] N. Ueda and R. Nakano. Deterministic annealing em algorithm. *Neural Networks*, 11(2):271 – 282, 1998.