Prediction and Ranking Algorithms for Event-Based Network Data

Joshua O'Madadhain (jmadden@ics.uci.edu)
Jon Hutchins (johutchi@uci.edu)
Padhraic Smyth (smyth@ics.uci.edu)
Department of Computer Science
University of California, Irvine

ABSTRACT

Event-based network data consists of sets of events over time, each of which may involve multiple entities. Examples include email traffic, telephone calls, and research publications (interpreted as co-authorship events). Traditional network analysis techniques, such as social network models, often aggregate the relational information from each event into a single static network. In contrast, in this paper we focus on the temporal nature of such data. In particular, we look at the problems of temporal link prediction and node ranking, and describe new methods that illustrate opportunities for data mining and machine learning techniques in this context. Experimental results are discussed for a large set of co-authorship events measured over multiple years, and a large corporate email data set spanning 21 months.

1. INTRODUCTION

Large data sets describing events over time and involving multiple participants are increasingly of interest from a data analysis perspective. Examples include:

- Email [6] (Enron: 250,000 emails, 28,000 people)
- Telephone calls [7] (AT&T: 275 million calls each day among 350 million individuals)
- Research publications interpreted as co-authorship events [13] (CiteSeer: 730,000 papers, 770,000 authors)

Social network analysis has, generally speaking, been applied to two types of data: persistent relationships (friendships, affiliations, web links, etc.) and discrete events (meetings, publications, communications, transactions, etc.). However, prior research on quantitative analysis methods for data of either type has largely focused on a static view of the data in which all links are considered simultaneously, even if the underlying data is known to change over time.

Event data is inherently temporal, with a time-stamp or fixed time interval associated with each event. As an example, consider Figures 1 and 2; each of these shows a method of visualizing the connections created between individuals by participating in events. Figure 1 shows a sequence of email messages between individuals A, B, C, D, and E; each vertical "timeslice" represents emails that were sent at the same time. Figure 2 shows a sequence of events (papers) and their participants (authors) as a hypergraph, where hyperedges represent papers (tagged by year) and vertices represent authors. These visualizations illustrate that there can be multiple different representations for event data: as a single aggregate network, a series of networks, and so on.

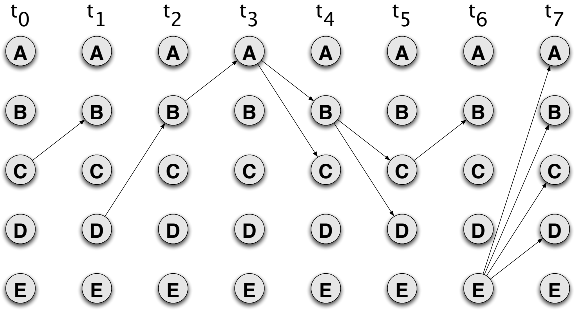

Figure 1: A sequence of events (emails).

From a data analysis and data mining perspective there are a number of different questions that can be asked in this context, including questions about how the networks evolve over time, emergence of communities, and so forth. In this paper we review our recent work on two specific problems related to event network data:

- **predicting future event co-participation of entities**: how likely is it that a given pair of individuals will co-participate in at least one event over some specific future time period? Specific examples of this question include "how likely is it that A will send an email to B in the next week?" and "How likely is it that author X will coauthor a paper with author Y next year?"
- **rank evolution**: how does the rank (prestige, influence, level of involvement, etc.) of each individual change over time in response to participation in a series of events? Answers to this question may be used to inform redistribution of resources (as certain individuals become more or less important/involved) or, more generally, to understand the evolution of the distribution of influence in any organization.

In Section 2 we describe some general notation for event networks and participants. In Section 3 we provide a brief review of relevant prior work in areas such as social network analysis, statistical network models, and machine learning.

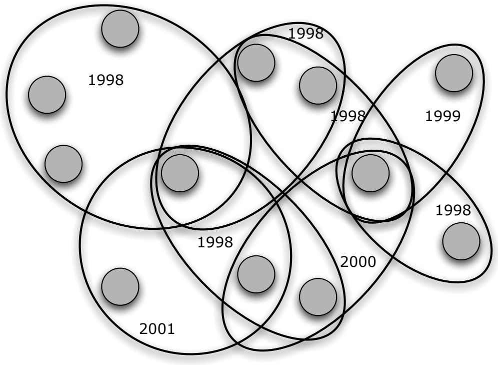

Figure 2: A co-authorship hypergraph tagged by year.

We then illustrate how machine learning techniques can be applied to the prediction problem described above and describe results from a set of experiments with a multi-year co-author data set consisting of 150,000 authors and 300,000 papers. In Section 4 we describe our work on algorithms for time-dependent ranking based on event data, and its application to a large corporate email data set. Section 5 provides concluding comments.

2. NOTATION

We define an event-based data set as consisting of a set of events $\mathcal{E} = \{e_1, \ldots, e_m\}$ and a set of participating entities $\mathcal{V} = \{v_1, \ldots, v_n\}$. The set of entities that participate in event e_i are denoted P_i, e.g., $P_1 = \{v_1, v_4, v_{10}\}$. Each event and each entity can have a set of attributes or covariates. We denote covariates for e_i as $\mathbf{y}_i$ and covariates for v_j as $\mathbf{x}_j$, and the sets of all event and entity covariates as $\mathbf{Y}$ and $\mathbf{X}$ respectively. For convenience, we denote the event covariate representing the time at which an event e_i occurred as t_i, where i indicates that it is the ith event in order of occurrence. Note that some events (such as emails) may have an "instantaneous" time-stamp t_i, while other events such as co-authorship events for technical papers would have a coarser time-stamp, e.g., at the level of year of publication, $t_i \in \{1985, 1986, \ldots, 2005\}$. Extensions of this representation could allow events to have a duration with a start and end time (such as meeting durations or tenure on a committee); however, we do not investigate the notion of events with duration in this paper.

Extending the notation above, we can express the proposition "v_j and v_k are co-participants in event e_i" by $v_j, v_k \in P_i$, and the proposition "v_j and v_k are co-participants in one or more events in the interval $[t, t+\Delta t]$" as $v_j, v_k \in P_{t,t+\Delta t}$. We denote the subset of events taking place in this interval as $\mathcal{E}_{t,t+\Delta t}$.

We denote the rank of v_j as $R(v_j)$, and (where appropriate) the rank of v_j at time t_i as $R_i(v_j)$. The notion of rank as an indicator of the significance of an entity is a key concept in algorithms such as PageRank, which we discuss further in Section 4.

Given this representation of the data, we can represent some of its structural properties as a network in various ways, depending on the precise nature of the events and of the desired analysis. Generally speaking, a network extracted from such a data set usually contains a vertex corresponding to each entity v_j, and edges link vertices that participate in the same events. However, as we will see, it is not always appropriate to aggregate all events together into a single graph for analysis.

3. MODELING AND PREDICTING LINK STRUCTURE

3.1 Statistical Models of Network Structure

Within the field of social networks there is a rich tradition of defining statistical models for network data. Various forms of Markov random fields (MRFs) [8] and exponential random graph models (sometimes referred to as p^* models in the social networks literature [27]) have been used to construct distributions $P(E)$ over the edge set E of a graph, or conditional distributions $P(E|\mathbf{X})$ given node covariates $\mathbf{X}$. Much of this work builds on the earlier classic work of Besag in spatial statistics [2].

The general goal is to infer a parsimonious model for $P(E)$ or $P(E|\mathbf{X})$ that requires a relatively small number of parameters to explain the pattern of observed relations (and non-relations), as a function of both local network properties (such as the indegree and outdegree of individual nodes) as well as the covariates $\mathbf{X}$. These generative modeling frameworks inherit the usual advantages of statistical modeling, including the ability to fit such models to data using statistical inference techniques, modeling techniques for incorporating covariates $\mathbf{X}$ (e.g., via suitably-defined logistic regression models [10]), inference methods for handling systematic errors in the measurements of links [5], hierarchical Bayes and random effect frameworks that allow individual-level variation to be modeled [9], and methodologies for incorporating specific prior information such as desired functional forms on degree distributions [23].

A major limitation of many of these models is lack of scalability as a function of the number of vertices. For example, in the latent variable model of Hoff, Raftery, and Handcock [10] the likelihood is by definition a product over all pairs of nodes, whether an edge was observed or not, leading to an inherently $O(n^2)$ algorithm. While this may be practical for relatively small social networks, such algorithms are not directly scalable to many of the large networks (in which $n \geq 10^5$) that are often of interest in data mining.

Work in statistical relational learning has also addressed the problem of building general-purpose statistical models of relational information. For example, Taskar et al. [26] use a relational Markov network (RMN) to define a probabilistic model over the entire network, including entity attributes and links. The primary goal of this work is to classify links by type, but the approach can also be applied to predicting link presence. Scalability to large networks is again an issue with such approaches.

In contrast to the approaches above, in Section 3.3 we describe how we construct local conditional probability models for link prediction in a manner that does not require the construction of a full joint distribution $P(E)$ over all edges. We do this by embedding the local graph structure and covariates in a fixed-dimensional feature space, allowing us to use standard predictive modeling techniques for learning conditional distributions, and to avoid the complexity of specifying joint distributions over sets of edges. While this loses some of the power of the full joint modeling approach, we will nonetheless argue that this can be an effective (and scalable) approach when prediction (in the form of queries regarding specified entity pairs) is the primary goal.

3.2 Modeling Networks over Time and Link Prediction

The work described in the previous subsection describes traditional modeling efforts that are largely focused on static networks. As we stated in the introduction, for event-based networks it is of interest to directly take the temporal and sequential aspect of the data into account. Below we summarize some recent work along these lines.

Perhaps the most widely-publicized work on modeling of temporal aspects of network data focuses on finding general-purpose stochastic laws or rules for link generation over time. Typically these laws or rules are governed by a small number of free parameters that control (for example) the probability of new link generation when a new node is introduced to the network. Examples of this approach include the preferential attachment model [16] and the "forest fire" model [14]. When used to simulate network data, these models yield networks with aggregate properties that are often quite similar to those seen in real-world network data, such as degree distributions, community characteristics, and network diameters. While these models have had considerable success in reflecting global aggregate network properties, they do not explicitly allow one to make predictions at the individual node or link level. For example, these models do not allow one to predict whether a new event will occur (over some future time span) involving two specific entities v_j and v_k, as a function of existing graph and covariate data pertaining to v_j and v_k.

Snijders [24] describes several interesting statistical models for understanding the dynamics of evolving networks of persistent relationships. These take as input sequential instantaneous observations of a network, and attempt to fit a specific parametric model for continuous evolution of the network such that its outputs (in the form of network "snapshots") accurately predict the observations. These methods are not directly applicable to event data, and as with the static modeling counterparts discussed earlier, parameter estimation for these models typically does not scale well to large networks.

Popescul and Ungar [19] present a method which involves performing a constrained search, in the space of database queries, to generate candidate features which are then used in a logistic regression to predict the presence or absence of a link between a specified pair of entities. Features are included or not based on a Bayes Information Criterion [21] evaluation. While this method was applied only to static graphs, it could in theory be extended to predict links in future time periods. However, the representation of the features as SQL database queries limits their expressiveness, and thus this method of feature generation cannot discover features which have been demonstrated to be highly informative. For example, SQL cannot readily represent features that relate to complex properties of the network topology, such as shortest-path distance in a network, or properties of an entity's neighborhood whose components are weighted according to their own topological properties ([1], [18], etc.). Likewise, features such as similarity metrics of entity or event attributes cannot be expressed in SQL.

Liben-Nowell and Kleinberg [15] rank all pairs of entities according to their value for a specified single network-based feature based on known event data, and declare that the k pairs with the highest feature value are those that will participate in an event in the following time period (where k is the number of pairs which are assumed to co-participate). This method does not scale well to large networks, involves some potentially problematic assumptions (such as prior knowledge of k), and may have low predictive accuracy since only a single feature is being used for event prediction.

3.3 Learning to Predict Co-Participation Over Time

We consider below the specific problem of answering the question "given the existing event data, will entities v_j and v_k co-participate in at least one event in a future specified interval?". Our approach is to treat it as a data-driven classification problem (in which "co-participating" is one class, and "not co-participating" is the other). The methods used are primarily probabilistic classifiers, which assign a probability to each class conditioned on the values of a set of specified features, whose nature may vary depending on the data set. We formally define this conditional probability as follows:

$$p(v_j, v_k \in P_{t,t+\Delta t} | f(\mathcal{E}_{1,t}, \mathcal{V}, \mathbf{X}, \mathbf{Y}) = \mathbf{w}) \qquad (1)$$

where $v_j, v_k \in P_{t,t+\Delta t}$ is a binary proposition defining whether entities v_j and v_k co-participate in any event in the time period $[t, t+\Delta t]$, f is a function returning a vector $\mathbf{w}$ of feature values, $\mathcal{E}_{1,t}$ is the historical event data up to time t, and $\mathbf{X}, \mathbf{Y}$ are the relevant entity and event covariate data.

This formulation frames the problem as one of learning a mapping from feature vectors to class probabilities; this problem is well understood by the machine learning community, and can be solved using standard "off-the-shelf" prediction algorithms.

There exist two variants of this problem: one in which v_j and v_k may or may not have co-participated in any previous interval, and one in which it is guaranteed that they have never previously co-participated (i.e., predicting new collaborations). The latter problem is generally considered more difficult, and is the one which we discuss below.

The primary components of a classification model are the choices of features, training and test sets, classification method, and evaluation metric; we will briefly discuss each of these below.

- **training and test sets**: we define the training period as the interval on which the training features are measured (e.g., years 1980 to 2003); the training "targets" (class labels) are defined on the time period following the training features (e.g., year 2004). Thus, the training data is constructed such that the model learns to map feature values from a time-interval to class labels in a future time-period; this is a standard approach in time-series modeling. The test data is constructed in a similar manner, but shifted in time, e.g., test features would be defined on years 1981 to 2004, and the class labels for the test data set would be defined on year 2005. (See Figure 3 for an illustration of the re-

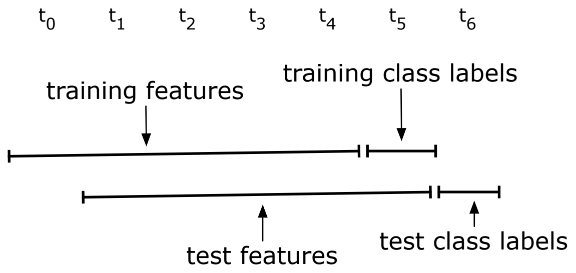

Figure 3: The intervals used for defining the training and test data

lationships.) Although there is overlap in the training and test feature sets, this setup matches exactly how training and prediction would be done in a real-world situation. For example, given event data through 2004, one could learn the relation between events in 2004 and features measured on events prior to 2004; this model could then be shifted in time to predict events in 2005.

- **features**: as already noted, the choice of features will depend on the data being analyzed, but in general, the features will be defined with respect to pairs of entities in the network. The features may be divided into two categories: network-based and entity-based. Network-based features include neighborhood-based features (such as Jaccard coefficient [20], Adamic/Adar coefficient [1], and weighted neighborhood cosine [18]) and the length of the shortest weighted path in the network. Entity-based features include such things as various entity attribute similarity measures, (for example, the KL-divergence of two entities' topic distributions [25] for data sets that include text), geographic proximity, and similarity of journal publication patterns.
- **classifier**: in general any classification method can be used for this problem. In practice we have found that the relatively simple approach of logistic regression seems to work well in that it is stable, interpretable, computationally efficient, and produces well-calibrated probabilities that are useful for ranking.
- **evaluation**: many event-based networks of interest are very large, and generally very sparse, i.e., the vast majority of entity pairs never co-participate. Thus, it is difficult to improve on always predicting the majority class (that none of the entities will co-participate in events in the future time period). Measuring the quality of a ranking produced by a classifier (e.g., receiver-operating characteristics, also known as ROC curves) produces more sensitive measures of classifier performance for this problem, such as the fraction of the k top-ranked future co-participations (as predicted by the classifier) that actually occur.

In large-scale network analysis, it is also important to consider the ramifications of different representations of the underlying data. In particular, while graphs are by far the most common representations of this type of data, they are not always the most appropriate. For example, if events involve multiple entities (such as coauthorship of a paper, or attendance at a meeting), then a graph-based representation will represent each entity as a vertex, and each event with k participants as a set of $O(k^2)$ edges (specifically, a clique of size k on the participants' vertices). This representation may require a great deal of space (for one data set, 150,000 events were represented by 2.2 million edges), tends to blur or erase the correspondence between the original data and the network topology (since each event in the folded graph is represented by a set of edges, each of which may be associated with > 1 event), and generally makes dealing with event metadata more difficult.

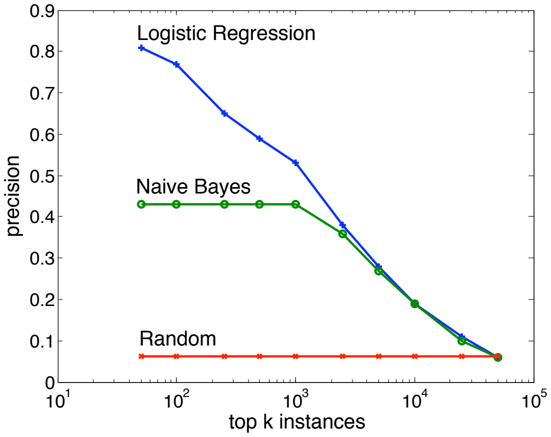

Figure 4: Precision of the top k instances, for three ranking methods

A more natural representation of such data is a hypergraph in which each entity is represented by a vertex, and each event is represented by a single hyperedge which connects the k entities that participated in the event (in contrast with the $O(k^2)$ edges required by the graph representation). This representation is much more efficient and accurate than the graph representation, but due to a lack of available programming tools that can directly manipulate hypergraphs, researchers in network analysis have generally chosen to accept the limitations of the graph representation. In the experiments described below we used the JUNG network software library [11], which can represent and manipulate hypergraphs as well as graphs; this allowed us to use the generally more advantageous hypergraph representation.

We have experimented with the approach of using classification methods to perform link (*i.e.*, event co-participation) prediction over time on a data set that consists of 128,000 scientific abstracts and the 310,000 authors of these publications, which spans the 6-year time period 1998-2003. Each paper constitutes an event, and all of the authors of the paper are co-participants in this event.

An example of a prediction problem for this type of data is to select the 1% most prolific authors (3100 of them) and try to predict new collaborations among these authors in the final year of the data. In the test set of 50,000 pairs of authors, approximately 6% of these pairs had new collaborations in the final year.

Figure 4 shows the precision for various values of k, defined as the fraction of the top-ranked k instances (from different models) that represent (true) positive instances (that is, pairs of individuals that were correctly predicted to co-participate in the specified time period). This fraction is an indicator of the utility of the probability returned by the classifiers in terms of ranking positive instances high and negative instances low; for example, we can see that of the 50 highest-ranked pairs according to the logistic regression method, 42 were positive instances. These results suggest that relatively standard machine learning methods can extract predictive power from this type of event data; in particular, this approach appears to be particularly useful as a ranking mechanism for detecting candidate pairs that are highly likely to co-participate in future events.

4. EVENT-BASED RANKING

We now switch attention to the second problem mentioned in the introduction, namely entity ranking from event-based data. There exist a variety of algorithms that rank entities in a network according to criteria that reflect structural properties of the network (such as the extent to which paths in the network pass through each individual); these rankings are interpreted as such qualities as "centrality", "authority", "influence", and so forth. Examples of these algorithms include betweenness centrality [3], eigenvector centrality [22], PageRank [4], and HITS [12]. Each of these algorithms makes the implicit assumption that the network data is static, and generates a single rank value for the data set; their underlying models do not incorporate any notion of sequence or timing, and all data are incorporated into a single picture of the network. While this assumption may be reasonable for networks based on persistent relationships (such as web page hyperlinks), it is less appropriate for event-based networks. For example, researchers gain prestige as their papers are cited, and lose it if they are no longer actively cited.

One can use such algorithms to generate ranks that change over time by applying them to subsets of the data restricted to successive intervals. However, this presents a few difficulties: even the subsets will be static pictures, and any information about the sequence of events during each interval is lost. More fundamentally, these algorithms necessarily operate on networks which represent aggregations of past and current events; it is neither useful nor especially meaningful for PageRank to operate on a network which contains many vertices but only a few edges corresponding to the current event(s). As such, the presence of links representing past events can cause the ranks to evolve in nonintuitive ways when small perturbations, corresponding to new events, occur. Furthermore, it is not clear what the semantics of the links induced by events should be for purposes of calculating an evolving ranking. For example, how should email events be represented by edges? Options include edges directed towards the sender, edges directed towards the receiver, and hyperedges connecting all participants. However, it is not clear how each of these choices would affect the ranks which emerge from PageRank and other algorithms which operate on this sort of data.

We illustrate one of these problems with the following example. Figure 5 is a network representation of email traffic, in which $< X, Y >$ exists if X has emailed Y, and has weight equal to the number of such emails. Any ranking algorithm which operates on static networks would assign the same rank to A and C. However, if we consider individual emails in sequence, we observe that patterns of communication may change over time, and thus the ranks may change as well: Figure 6 represents a possible sequence of messages that correspond to the static network in Figure 5. In this representation C at first is more important than A, and later this reverses. Furthermore, participation in an event at time t_i can affect one's participation in events at time $t_j (j > i)$ (for example, based on Figure 6 we can guess that D's email to B at time t_4 may have resulted in B's email to A at time t_5, but the reverse is clearly impossible). These observations underline the desirability for an event-based ranking algorithm to be able to generate ranks which evolve over time, and whose ranks respect the events' temporal sequence.

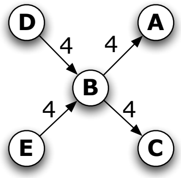

Figure 5: A network representing a sequence of messages

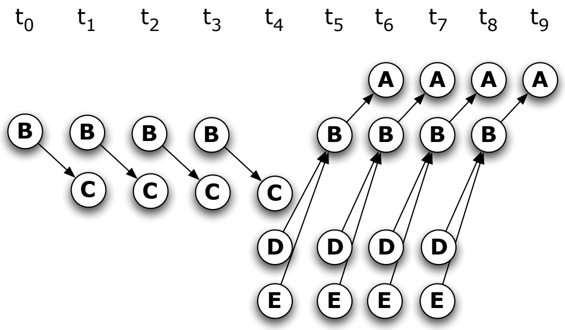

Figure 6: A message sequence that could have resulted in the network in Figure 5

We have identified several properties which we believe ought to be satisfied by algorithms which generate a sequence of values which model the evolution of ranks over time based on event participation:

1. comparability across time: rank values should be normalized so that the ranks at time t_i and t_j may be directly compared
2. participation increases rank in proportion to other participants' rank
3. participating can always increase rank (even if participants' ranks are all 0)
4. participants' ranks don't decrease
5. non-participants' ranks don't increase
6. rank value evolution reflects event sequence

As already noted above, PageRank and similar algorithms do not satisfy (4), (5), and (6) in general, and may not satisfy (3) in some cases.

The functioning of PageRank and related algorithms can be modeled in terms of iterated "potential" flow, according to the following specifications:

- **initialization**: each individual starts out with an equal amount of potential
- **iteration**: the algorithm iterates until all individuals' net change in potential is zero; the number of iterations will depend on the initial distribution of potential and the topology of the network
- **flow paths**: potential flows from each individual to its neighbors in the network (in the case of PageRank, a constant fraction of the total potential is "held out" and distributed equally among all individuals)

EventRank [17] is a framework for ranking algorithms which operate on event data and incorporate temporal information. Algorithms from this framework can also be conceptualized in terms of iterated potential flow. However, while the initialization is identical to that of PageRank, the others differ as follows:

- **iteration**: each event causes exactly one iteration to occur
- **flow paths**: potential flows from non-participants (in the current event) to participants; past events are not considered except insofar as they are manifested in the previous rank values

We define the basic model as follows: we denote the potential of participant $v \in V$ at time t_i by $R_i(v)$, which takes on values in the interval $(0, 1)$. $R_0(v) \equiv \frac{1}{n}$ (uniform initial distribution), and in general $R_i(v)$ is recursively defined as

$$v \in P_i \quad : \quad R_{i-1}(v) + \alpha_i \cdot \frac{\bar{R}_{i-1}(v)}{\sum\limits_{d \in P_i} \bar{R}_{i-1}(d)} \tag{2}$$

$$v \notin P_i \quad : \quad R_{i-1}(v) \cdot \left(1 - \frac{\alpha_i}{T_{N_{i-1}}}\right) \tag{3}$$

where α_i is the total amount of potential that the event e_i contributes to the participant set, $\bar{R}(d, t_i)$ is the additive inverse of d's potential, i.e., $1 - R_i(d)$, and $T_{N_{i-1}}$ denotes the total amount of potential held by the non-participants of m_i, that is, $\sum\limits_{d \notin P_i} R_{i-1}(d)$.

Further details of the model may be found in [17], in which we argue that this model satisfies all of the requirements stated above.

As a test of this framework, we performed experiments on approximately 1 million emails spanning 21 months of an organization's email server log, for 628 individuals (Figure 7). The data for each message included the identities of the sender and recipients, and the time at which it was sent, but not the content of the message itself. We also had access to a single snapshot of the organizational hierarchy for 378 members of the organization.

The definition of rank in a social network is generally somewhat subjective; ranking algorithms generally do not have a "ground truth" to which their output can be compared to determine accuracy of the ranking model, although for smaller social networks one can measure the consistency of algorithmically determined ranks with those derived from individual surveys. While we have observed that EventRank satisfies the properties listed above, a more objective measure of relevance is desirable.

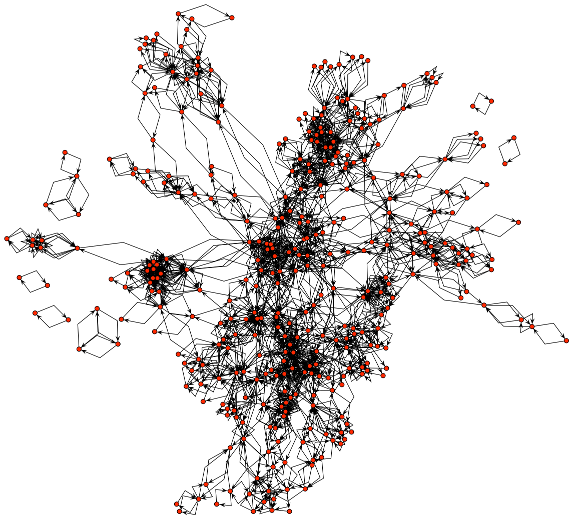

Figure 7: A static view of an organization's email network

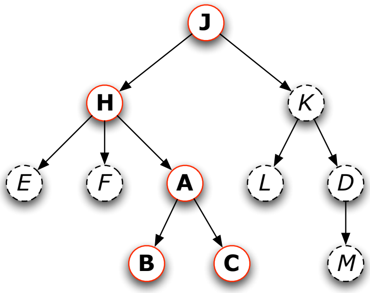

Figure 8: An example of an organizational hierarchy

We hypothesized that, in general, the rank of an individual is related to her position in the organizational hierarchy (see Figure 8 for an illustration), in the following senses: first, that an individual A's rank is correlated with the number of her subordinates, and second, that A's rank should be greater than that of those below her in the hierarchy (subordinates: B, C), and less than that of those above her in the hierarchy (superordinates: J, H).

Since the organizational hierarchy is itself a static network, for purposes of these comparisons we needed to define a single "cumulative" rank value for each individual. One of the measures that we chose was the sum of "incoming" potential (that is, changes to c's potential caused by c receiving a message), which we denoted as S_i; this is analogous to the HITS "authority" score, and to network indegree.

Generally, we observed that S_i had a weak linear correlation (0.47) with log(number of subordinates). We also measured the extent to which the rank ordering derived from S_i agreed with the hierarchy (in the sense defined above), and found that this ordering was highly consistent: the average number of inversions (situations in which a subordinate's rank was higher than that of a superordinate) was lower, and the overall accuracy higher, than that for any other rank ordering we tested. Additional details on these results may be found in [17].

While we did not have a direct way of validating Event-

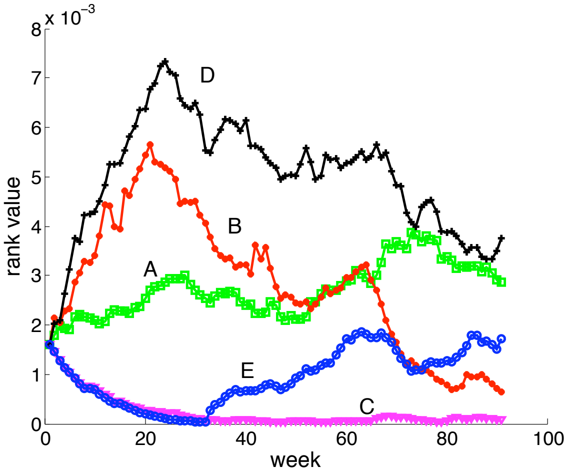

Figure 9: Rank vs. time for 5 individuals

Rank's time-varying ranks for each individual, we did examine the data for that of a few individuals about which some additional information was known. Figure 9 shows a plot of rank vs. time for five individuals. Individual *A* was working on projects of increasing visibility during this time period; *B* went on leave around week 60 or so; *C* worked primarily on their own rather than with other members of their group; *E* did not start working in this organization until approximately week 30; and *D* was the leader of the group which included the other four individuals. Intuitively, this time series seems to agree with this limited information.

5. CONCLUSION

We have presented methods for network analysis that explicitly incorporate time and sequence, and are thus well-suited to addressing event data sets. We argued that an emphasis on predictive modeling, using techniques from data mining and machine learning, can yield scalable and robust algorithms in this context. Experimental results demonstrated that these approaches can be used to accurately predict organizational structure from event data and to rank likely future co-participations between entities.

There are many other open research problems relating to event-based data sets and the evolution of their associated networks; we believe that such problems present opportunities for new practical applications and for a better understanding of the dynamics of the underlying phenomena.

Acknowledgements

This material is based upon work supported by the National Science Foundation under award number NSF IIS-0083489.

6. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, July 2003.

[2] J. Besag. Spatial interaction and the statistic analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, pages 192–293, 1974.

[3] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] C. Butts. Network inference, error, and informant (in)accuracy: A Bayesian approach. *Social Networks*, 25(2):103–140, 2003.

[6] W. W. Cohen. Enron email dataset. http://www.cs.cmu.edu/∼enron/, 2005.

[7] C. Cortes and D. Pregibon. Giga-mining. In *Knowledge Discovery and Data Mining*, pages 174–178, 1998.

[8] O. Frank and D. Strauss. Markov graphs. *Journal of the American Statistical Association*, pages 832–842, 1986.

[9] P. D. Hoff. Random effects models for network data. In R. Breiger, K. Carley, and P. Pattison, editors, *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, pages 303–312. The National Academies Press, 2003.

[10] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

[11] JUNG Framework Development Team. JUNG: The Java Universal Network/Graph Framework. http://jung.sourceforge.net.

[12] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[13] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and autonomous citation indexing. *Computer*, 32(6):67–71, 1999.

[14] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters, and possible explanations. In *Knowledge Discovery and Data Mining (KDD)*, Chicago, IL, August 2005. ACM SIGKDD.

[15] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Conference on Information and Knowledge Management (CIKM)*, 2003.

[16] M. Newman. Clustering and preferential attachment in growing networks. http://aps.arxiv.org/abs/cond-mat/0104209/, 2001.

[17] J. O'Madadhain and P. Smyth. EventRank: A framework for ranking time-varying networks. In *Third International Workshop on Link Discovery (LinkKDD'05)*, pages 9–16, Chicago, IL, August 2005. ACM SIGKDD.

[18] J. O'Madadhain, P. Smyth, and L. Adamic. Learning predictive models for link formation. Presented at the International Sunbelt Social Network Conference, 2005.

[19] A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI03 Workshop on Learning Statistical Models from Relational Data*, 2003.

[20] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[21] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978.

[22] J. R. Seeley. The net of reciprocal influence: A problem in treating sociometric data. *Canadian Journal of Psychology*, 3:234–240, 1949.

[23] T. A. Snijders. Accounting for degree distributions in empirical analysis of network dynamics. In R. Breiger, K. Carley, and P. Pattison, editors, *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, pages 146–161. The National Academies Press, 2003.

[24] T. A. Snijders. *Models and Methods in Social Network Analysis*, chapter 11, pages 215–247. Number 28 in Structural Analysis in the Social Sciences. Cambridge University Press, April 2005.

[25] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the Tenth ACM International Conference on Knowledge Discovery and Data Mining*, pages 306–315, Seattle, WA, 2004. ACM Press.

[26] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2003.

[27] S. Wasserman and P. Pattison. Logit models and logistic regression for social networks: I. An introduction to Markov graphs and $p*$. *Psychometrika*, pages 401–425, 1996.