EventRank: A Framework for Ranking Time-Varying Networks

Joshua O'Madadhain (jmadden@ics.uci.edu)
Department of Computer Science
University of California, Irvine

Padhraic Smyth (smyth@ics.uci.edu)
Department of Computer Science
University of California, Irvine

ABSTRACT
Node-ranking algorithms for (social) networks do not respect the sequence of events from which the network is constructed, but rather measure rank on the aggregation of all data. For data sets that relate to the flow of information (e.g., email), this loss of information can obscure the true relative importances of individuals in the network. We present EventRank, a framework for ranking algorithms that respect event sequences and provide a natural way of tracking changes in ranking over time. We compare the performance of a number of ranking algorithms using a large organizational data set consisting of approximately 1 million emails involving over 600 users, including an evaluation of how the email-based ranking correlates with known organizational hierarchy.

Categories and Subject Descriptors
H.2.8 [**Information Systems**]: Database applications—*data mining*; G.2.2 [**Discrete Mathematics**]: Graph theory—*network problems*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*sociology*

General Terms
Algorithms, Experimentation, Measurement, Verification

Keywords
Network ranking algorithms, network temporal evolution, social network analysis

1. INTRODUCTION AND MOTIVATION
There exist a variety of algorithms that rank entities in a social network according to criteria that reflect structural properties of the network. These criteria are generally intended to measure the "influence", "authority", "prestige", or "centrality" of the entities in the community represented by the network. Examples of such ranking algorithms include betweenness centrality [2], eigenvector centrality [9] (and related algorithms such as PageRank [3]), HITS [6], and voltage-based rankers [12].

LinkKDD'05 August 21, 2005, Chicago, IL, USA

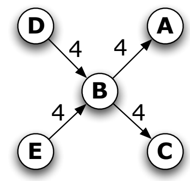

Figure 1: A network representing a sequence of messages

Each of these algorithms makes the implicit assumption that the network is a static object, i.e., that the entity and relationship sets, and the rank of each in the community, do not change over time. However, in some cases, this assumption is clearly false [4]. Examples include research citation networks (researchers may gain prestige over time if they publish papers that are cited by many people, and may lose it if they stop publishing), and email networks (correspondents' participation in email, and thus the extent to which they are "in the loop", may change on several time scales, depending on such factors as patterns of email access, vacations, and changes of status in an organization).

It is, of course, possible to repeatedly apply one of the above ranking algorithms to successive "snapshots" of the data (that is, subsets of the data restricted to a particular interval) to yield a sequence of rank values that vary over time [5]. However, in the context of data sets that relate to the flow of information (e.g., email), the sequence of events can be significant in determining the relative importances of individuals in the network, and this information is lost when events are aggregated into a single snapshot of the network. Thus, rank values on networks which represent an aggregation of data over time can be thought of as representing summary statistics (e.g., sums or means) of the ranks over time.

One can also, given a static picture of a network, assign weights to edges that reflect the amount of elapsed time since each associated event occurred [13]. However, while this may yield a better model of the edge weights at a given time than a simple summation of the number of events in which the individuals mutually participated, it still fails to capture the information represented by the sequence of messages.

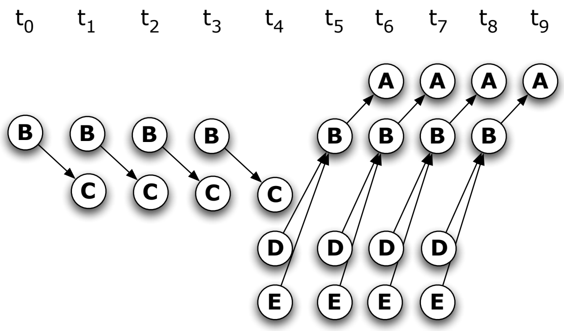

Figure 2: A message sequence that could have resulted in the network in Figure 1

For an illustration of this information loss, consider the example network shown in Figure 1, in which the directed edge $< X, Y >$ exists if X has emailed Y, and the value associated with $< X, Y >$ denotes the number of messages that X has sent to Y. Any of the ranking algorithms mentioned thus far would conclude that the ranks of A and C were the same; in the terminology of social network analysis, A and C are structurally equivalent.

However, the sequence in which email is sent carries information about the underlying communication of information; we can model correspondents as having a state, which changes in response to the receipt (and possibly the sending) of a message. Suppose that we have two emails e_{XY} from X to Y, and e_{YZ} from Y to Z; we denote the time of an email by $t(e_{UV})$. If $t(e_{XY}) < t(e_{YZ})$, then the content and timing of e_{YZ} may reflect the state change caused by e_{XY}. However, if $t(e_{XY}) > t(e_{YZ})$, then e_{YZ} cannot reflect any information contained in e_{XY}.

Figure 2 represents one possible sequence of messages that could have resulted in the collapsed network shown in Figure 1; there are, of course, many such sequences. This sequence suggests that, at time t_9, A should be considered to be more important than C, both because A has been a message participant more recently, and because the sequence suggests that A may be receiving information from D and E as well as from B. It is also important to point out that at time t_4 the opposite is true (that is, we would consider C to be more important than A).

We will discuss two different types of measures for temporal ranking: *transient*, which is a measure of the current rank at a particular time t, and *cumulative*, which is a measure of rank that encompasses the interval $[t_0, t]$.

In this paper, we will describe a framework for such measures. Our examples and model will focus specifically on email traffic data, but we believe that this framework may have wider application to ranking entities in data sets which consist of sequenced events that induce (or reflect) a network of relationships.

2. MODEL

We can model the functioning of algorithms such as PageRank or the voltage ranking algorithm as the flow of "potential" in a network from each entity to neighboring entities; possession and/or transfer of this potential is the basis for these measurements of rank. This potential flow can be modelled by repeated multiplication of the vector of original potential values (generally a uniform distribution) by a matrix M which represents the network.

We borrow this metaphor of potential flow to describe the functioning of the models in our framework: the potential values at time t_{i+1} may be calculated based on those at time t_i by multiplying the potential vector by a matrix M_i which represents the effect of the message sent at time t_i. Thus, transient rank may be defined as the amount of potential present at time t, whereas cumulative rank may be defined in terms of the mean potential value for the interval $[0, t]$.

There are two key distinctions between existing models and those arising from the framework that we propose. First, algorithms such as PageRank generate rankings that correspond to a stationary distribution of potential over entities, but by design, our models generate ranks that do not converge to a single value, because the matrices M_i, which represent messages with different senders and recipients, are not all identical. Second, algorithms such as PageRank use potential flow as a model of a random traversal of the network; by contrast, our models use potential flow to model exactly those transitions which correspond to the events for which we have evidence, in the sequence in which the events occurred: potential flows if and only if a message is sent.

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 ranks for smaller social networks can be validated in part by comparing their results with those of surveys of entities in the network. As such, the validity of a ranking model is generally evaluated first in terms of its axiomatic properties. While not quite a complete axiomization, we nonetheless present here a list of desiderata that we believe should be satisfied by any model whose purpose is to calculate entity ranks based on email traffic.

Note that in the list below, a message *participant* is either a sender or a recipient of a specific message m; all other entities are *non-participants* of m.

1. Ranks should be comparable across time. (Ideally this would mean that at each step that ranks are automatically normalized, but at least they should be able to be normalized at any time.)
2. Receiving a message from an individual of rank r should lead to an increase in rank at least as large as that from receiving a message from an individual of rank $q < r$, all other things being equal.
3. Sending a message should not fail to have an effect because the sender has no "potential".
4. The ranks of the participants of m should not decrease in response to m. (There might be circumstances in which sending and receipt should have no effect on participants' ranks.)

5. The ranks of the non-participants of m should not increase in response to m. (Again, it may be permissible for non-participants' ranks to remain constant.)

6. Rank value evolution should be sensitive to message sequence.

These requirements might appear more stringent than those of other ranking algorithms, but this is essentially a reflection of the fact that we wish to model the effect of individual successive events (in this case, emails) on rank values, as opposed to modeling the effects of all such events in parallel.

The potential flow for a message in this framework will take the following general form: the non-participants send some of their potential to the sender; the sender retains some fraction of this potential (which causes the sender's potential to increase) and distributes the rest among the recipients (which causes each of the recipients' potential to increase). This scheme satisfies each of the requirements enumerated above: potential is conserved, which means that the transient ranks are automatically normalized (and thus comparable across time); the sender always has potential to send (unless there are no non-participants, in which case the message is effectively spam and should have no effect on anyone's rank); the participants gain potential (or at least lose none); and the non-participants lose potential. If we process messages in chronological order, then models in this framework will automatically satisfy all our stated requirements.

We denote the potential of correspondent $c \in C$ at time t_i by $R_i(c)$, which takes on values in the interval $(0, 1)$. $R_0(c) \equiv \frac{1}{|C|}$, and in general $R_i(c)$ is recursively defined as

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

$$c \notin P_i \quad : \quad R_{i-1}(c) \cdot \left(1 - \frac{\alpha_i}{T_{N_{i-1}}}\right) \tag{2}$$

where m_i is the message sent at time i, P_i is the set of participants of message m_i, α_i is the total amount of potential that the message m_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, $\sum\limits_{d \notin P_i} R_{i-1}(d)$.

The α_i values characterize the potential values' volatility–that is, larger values indicate that non-participants retain less of their potential–and are constrained as follows:

$$0 \leq \alpha_i \leq T_{N_{i-1}} \tag{3}$$

The definition of α_i may depend on a number of factors, such as the size of P_i relative to $|C|$, the elapsed time since the most recent message that the sender received from any of the recipients, the number of messages that the recipients have sent to the sender for which replies are pending, the elapsed time since the most recent message that the sender has sent, and so forth. Note that if $\alpha_i = 0$, then the potential values do not change at time step t_i, and if $\alpha_i = T_{N_{i-1}}$, then the potential values of the non-participants go to 0 (and further transfer of potential in response to subsequent messages will not occur as long as their participant sets are equal to P_i).

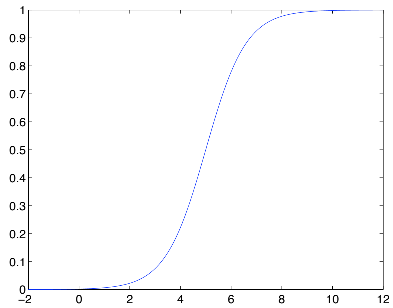

Figure 3: $g(\Delta t_s, 5)$

We observe that, in general, as P_i grows, the total amount of potential available ($T_{N_{i-1}}$) decreases; also, the changes in potential value to $d \in P_i$ decrease (because α_i is smaller, and because the number of correspondents among which α_i is divided is larger). In particular, if $P_i = C$, then m_i does not result in any transfer of potential. This ensures that messages with wide distribution ("spam") have little or no effect on potential.

The potential α_i is distributed among the elements of P_i in proportion to the additive inverse of their potential values: thus, the lower the potential of a participant, the more potential is assigned to it. (Senders and recipients, in this portion of the model, are treated equivalently; a more complex model might give senders or recipients more "credit" for their participation.)

In this paper, we explored two models for defining α_i. In the first ("baseline"), α_i was set to a constant fraction $f \in (0, 1)$ of $T_{N_{i-1}}$ for various values of f:

$$\alpha_i = f \cdot T_{N_{i-1}} \tag{4}$$

The second ("reply") model for α_i, elaborates the baseline model for α_i by discounting it according to functions of two factors: Δt_s (the elapsed time since the last message sent by the sender of m_i), and Δt_r (the elapsed time since the last message received by the sender from any recipient of m_i):

$$\alpha_i = f \cdot T_{N_{i-1}} \cdot g(\Delta t_s, G) \cdot h(\Delta t_r, H) \tag{5}$$

where g and h both take on values in the interval $(0, 1)$.

We define g as

$$g(\Delta t_s, G) = \frac{\tanh(\frac{10 \Delta t_s}{G \cdot \pi} - \pi) + 1}{2} \tag{6}$$

where G is a positive constant that specifies the amount of time required for a sender to "recharge" to the point that her next message will have half the maximum possible effect. Figure 3 shows that this functional form for g guarantees that its output increases with Δt_s, while being restricted to the range $(0, 1)$. This modification is motivated by our desire to prevent individuals who send messages much more frequently than the norm (specified by G) from dominating the rankings.

We define h as

$$h(\Delta t_r, H) = 2^{-\frac{\Delta t_r}{H}} \tag{7}$$

where H is a positive constant that specifies the "half-life" of a message (the amount of time required for the effect of a reply to drop to half the maximum); this boosts α_i for messages that are quick replies to other messages. (Since we do not have access to the email headers in the experimental data used below, we assume that a message sent from s to a set of recipients D is a "reply" if any $c \in C$ has sent a message to s since the last message that s sent to D.) This refinement to the model reflects observations that have been made to the effect that the speed of response to email can carry information [11].

It is possible to model G and H as functions of the characteristics of individual participants, or of pairs of individuals, rather than as global features; H, in particular, is likely to depend in practice on individuals' attitudes regarding email etiquette. However, we do not have enough information for the data set used in this paper to be able to model individual characteristics in this way.

We note that there are many other plausible approaches to defining a set of update equations and parameters and we do not claim that the specific methods proposed above are in any way unique or optimal. Later for example we will look at the sensitivity of the results to whether or not a reply component is included in the model, the setting of the parameter f, and so forth.

The time complexity of handling a single message is nominally $O(n)$; however, we can increase the efficiency by lazily updating the potential values of non-participants (that is, only updating the potential values of P_i at time t_i). We do this by storing (a) the sequence of α_i values, and (b) for each correspondent c, the index of the last message for which c was a participant. We then can apply all "skipped" α_i values at once when the next message for which c is a participant is processed (or after all messages have been processed, if there is no such message).

Based on this model, we define the following measures of cumulative rank for a correspondent c: sum of "outgoing" potential (that is, changes to c's potential caused by c sending a message), sum of "incoming" potential, and sum of transient ranks (that is, sum of potential values at each step); we refer to these hereafter as S_o, S_i, and S_r respectively. Note that S_o and S_i are analogous to the HITS "hub" and "authority" scores, respectively, or to outdegree and indegree.

3. EXPERIMENTS

Our experiments were performed on approximately 1 million emails spanning 21 months of an organization's email server log, for 628 individuals. Emails to and from extra-organizational entities were removed, as were all "broadcast" emails. The server log data for each message included a message ID, the identities of the sender and recipients, and the time at which it was sent. For each message, the sender was removed from the set of recipients, if it was present. We also had access to the organizational hierarchy for 378 members of the organization, so we could calculate both the depth of each individual in the hierarchy (their distance from the top) and the number of subordinates that they supervised. We did not have access to the content or the message headers; thus, we knew neither how much (original) information a message m_i contained, its similarity or relations to other messages that the participants may have sent or received, nor whether it was in fact a reply to an existing message.

For privacy reasons, we do not refer to specific individuals in this data set by name.

We tested various values of f ($\{0.001, 0.01, 0.1, 0.9\}$) for both models for α_i. For the second model we set H to 1 day, based on observations made in [11], and set G to 1 hour.

We performed three separate sets of experiments: measuring the relations between our algorithms' rank ordering and properties of the organizational hierarchy, comparing our algorithms to others on the basis of their fidelity to the organizational hierarchy, and measuring the sensitivity of our algorithms' performance to parameter values.

The experimental code was written in Java, using the JUNG [8] libraries for network representation and analysis; some of the post-analysis used MATLAB. A single model instance required approximately 8 minutes to process the messages on a dual 2.5 GHz Apple PowerMac with 2.5 GB RAM (1.5 GB of which was allocated to Java heap space).

4. RESULTS AND DISCUSSION

As previously observed, there exists no ground truth to which we can compare the results of our ranking algorithms to determine their correctness. This is particularly true for the transient rank measurements ($R_i(c)$), since network ranking algorithms are generally applied to static networks representing all data. For this reason, evaluation of our models focused primarily on the derived cumulative rank measurements defined earlier: S_i, S_o, and S_r.

We represented the organizational hierarchy as a tree in which A is a child of B iff A is supervised by B; we then defined the *depth* in the hierarchy to be the number of steps from the root of the tree (that is, the person in charge of the organization), A's *subordinates* as the individuals in the subtree rooted at A (not counting A itself), and A's *superordinates* as the individuals on the path from A to the root of the hierarchy.

We derived ranks for HITS (hub score and authority score), PageRank (with random restart probability α of 0.1), and weighted indegree and outdegree by applying them to the network where X is connected to Y if X has ever emailed Y. We defined the weight of an edge $< X, Y >$ to be the sum of the weights of each message from X to Y (normalized appropriately for HITS and PageRank); the weight of a message was in turn defined as (a) proportional to $\frac{1}{|P_i|}$ (for HITS and PageRank) or (b) 1 (for weighted in- or outdegree, i.e., each individual is ranked according to the number of messages sent or received).

The following analyses focus primarily on the HITS authority score, PageRank, indegree, and S_i and S_r (sum of incoming potential and sum of transient ranks) models for ranking. These "inflow"-based and direction-agnostic ranking methods generally performed much better in this context than their "outflow"-based counterparts, so for reasons of space the "outflow"-based results are largely omitted.

4.1 Rank and organizational hierarchy

Figure 4 shows the results of plotting the measured rank (from 0 to 627) against the hierarchy depth for several different ranking algorithms, where the distribution of ranks are presented as a box-plot[1] for each depth and each panel illustrates the results for a different ranking method. We observe the following:

- All of the ranking methods show that rank as determined from email is strongly dependent on tree depth in the organizational hierarchy: individuals who are highly ranked in the email data tend to be near the root node of the organizational tree, and vice-versa.

- The HITS authority ranking is the only one where the median rank does not monotonically increase with tree depth (there are 2 reversals). All of the other ranking methods are monotone in this sense–indeed their ranks are all strongly correlated with each other, while the ranks of HITS authority are much less correlated.

- It should be noted that tree-depth is not necessarily by itself a good predictor of importance in an organization. For example, at depth 2 in the tree are individuals who are likely to be administrative assistants or advisors to the individual at the root node, but who have no subordinates. We can see the existence of such individuals as outliers in several of the ranking plots for depth 2. (Below we look at a more subtle measure of organizational importance, namely the number of subordinates for each individual in the tree).

Figure 5 shows the results of plotting the rank values against the number of subordinates (for those with ≥ 1 subordinates) for the same ranking algorithms. We found that using the log of the number of subordinates produced a more interpretable plot compared to use the number of subordinates directly (which resulted in a very skewed plot)—in addition, the correlation between rank values and log-subordinates (for different methods) was significantly higher than for subordinates directly.

We see in Figure 5 a clear dependence between rank values and log(number of subordinates), with the exception again of HITS (authority ranking). In fact the dependence is weakly linear, as the correlation coefficients indicate (hovering around 0.4 and 0.5 for the higher correlations). Again, the InDegree, PageRank, baseline, and reply models were all highly correlated with each other with correlation coefficients of 0.8 and above (not shown).

Figure 5 confirms the results using tree-depth earlier: ranks based on email traffic are strongly correlated with the number of subordinates, at least for this data set. This is not particularly surprising, but nonetheless is informative to see borne out in practice. Outliers in these plots could for example be examined to see who they correspond to in the organization, e.g., employees who are low in the organization tree but who are ranked highly based on email traffic, or vice-versa.

[1]The box summarizes the distribution of a value: the horizontal lines show the lower quartile, median, and upper quartile values; the vertical lines show the extent of the data; outliers are plotted separately[10].

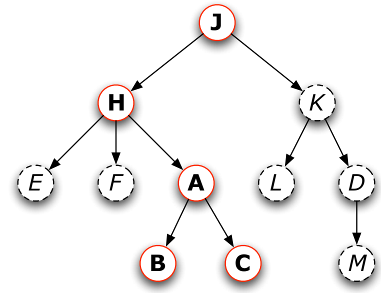

Figure 6: A sample hierarchy, with A's subordinates and superordinates highlighted.

4.2 Comparing rank algorithms

We hypothesize that an individual A should in general have higher rank than her subordinates and lower rank than her superordinates in the hierarchy tree. (We do not compare an individual to others than her sub- or superordinates, since it is not obvious that any consistent relationship ought to obtain between them.) Figure 6 highlights the subordinates (B, C) and superordinates (H, J) of A in a sample hierarchy tree.

On this basis, we say that, for a given algorithm, A is *inverted* with respect to its superordinate H if A's rank is higher than H's, and inverted with respect to its subordinate B if A's rank is lower than B's. We can then compare the performance of different ranking algorithms to one another based on measuring the inversions induced by each; an algorithm that corresponded perfectly to the hierarchy would have no inversions. Note that in general there are many different such rankings for a given tree; for instance, one can swap the ranks of sibling leaves (in Figure 6: B and C, or E and F) without affecting the number of inversions.

We can derive error measures from these inversions for each individual c in a few different ways: a simple summation of inversions, which we denote by $I(c)$; a weighted sum of inversions, based on rank difference or on depth difference (in which an inversion counts more if the rank/depth difference is greater), which we denote by $I_R(c)$ and $I_D(c)$ respectively; or a normalized count $I_N(c)$. I_N takes on values in $[0, 1]$, where 0 indicates no inversions and 1 indicates that all sub- and superordinates of c are inverted with respect to c); the additive inverse of this value is can be interpreted as an accuracy score. Note that I, I_R, and I_D place more emphasis on individuals with many subordinates, while I_N weights each individual equally.

Table 1 shows the result of calculating mean values for I, I_R, and I_N. (I_R and I_D turn out to be strongly correlated for this data set, so for simplicity we do not include figures for I_D here.) We observe the following:

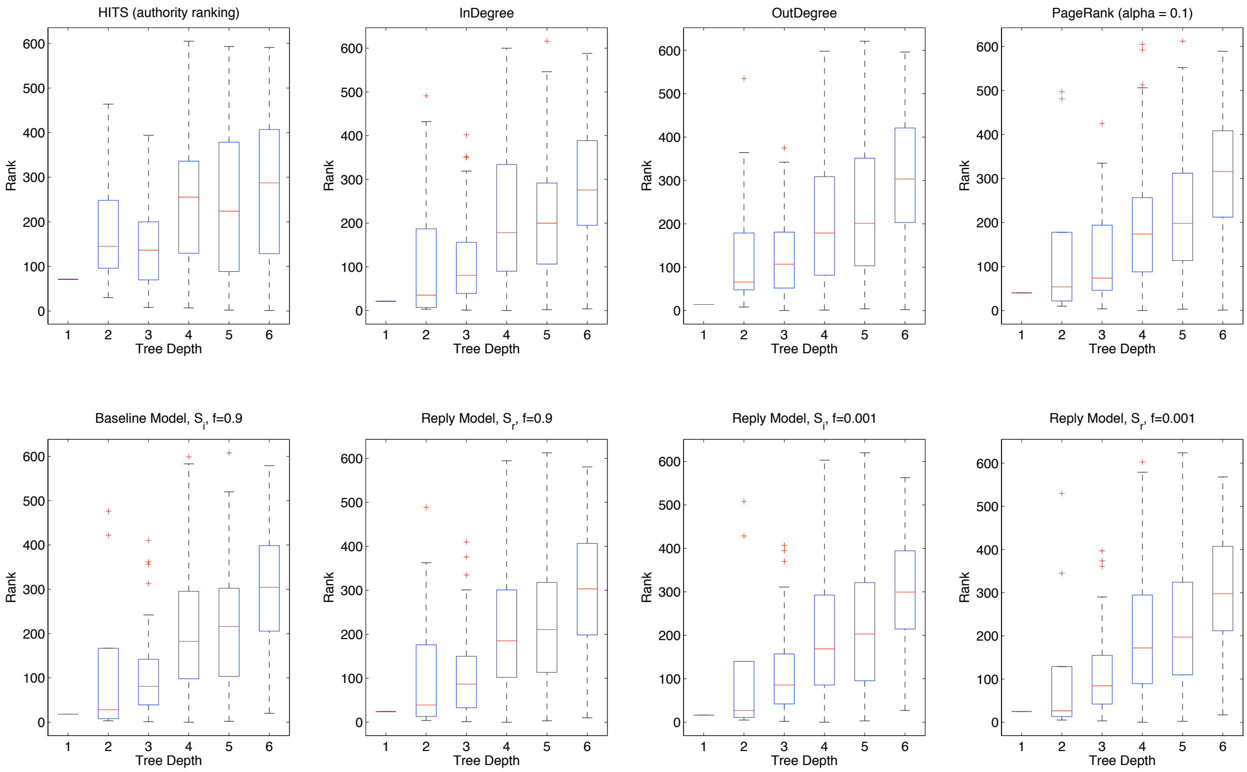

Figure 4: Rank versus depth for HITS (authority score), indegree, outdegree, PageRank, baseline model ($f = 0.9, S_i, S_r$), reply model ($f = 0.001, S_i, S_r$)

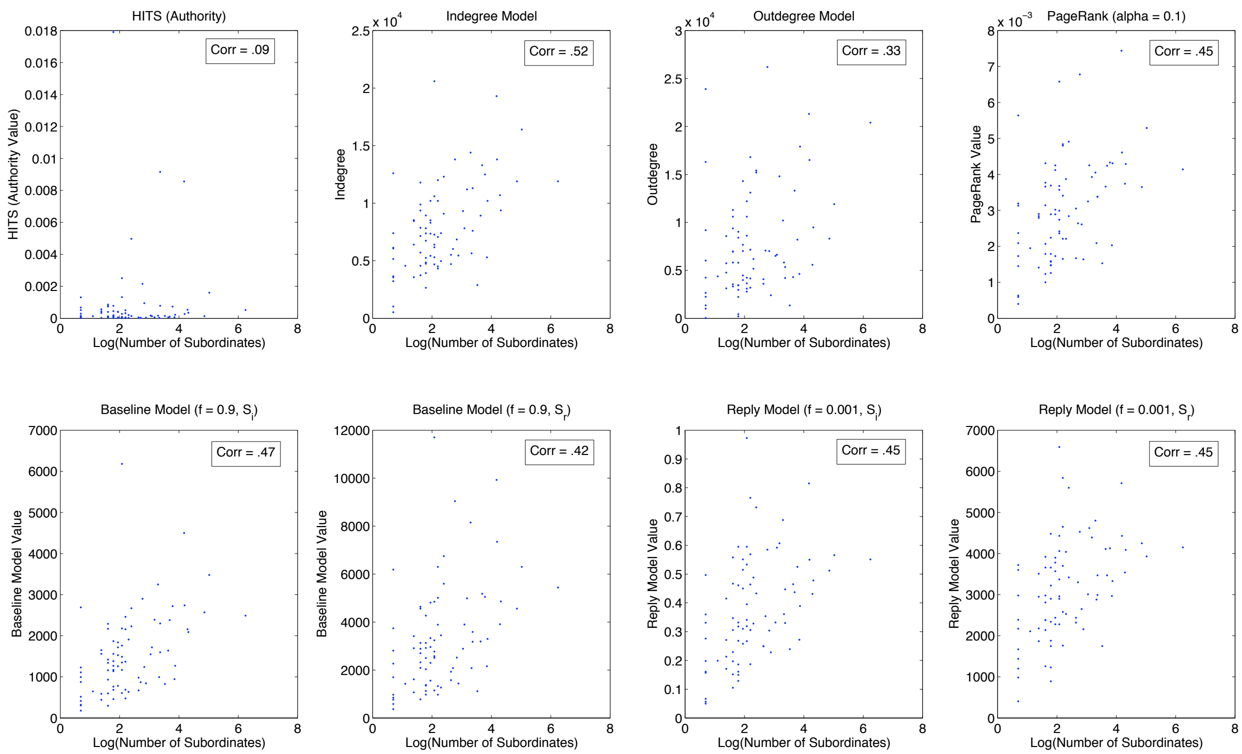

Figure 5: Rank value versus number of subordinates for HITS (authority score), indegree, outdegree, PageRank, baseline model ($f = 0.9, S_i, S_r$), reply model ($f = 0.001, S_i, S_r$)

	$\bar{\mathbf{I}}$	$\bar{\mathbf{I}}_{\mathbf{R}}$	$\bar{\mathbf{I}}_{\mathbf{N}}$
HITS (authority)	1.17	48.88	0.88
PageRank ($\alpha = 0.1$)	0.80	41.14	0.92
indegree	0.54	39.38	0.95
baseline, S_i ($f = 0.9$)	0.41	16.05	0.96
baseline, S_r ($f = 0.9$)	0.63	33.33	0.94
reply, S_i ($f = 0.001$)	0.50	18.94	0.96
reply, S_r ($f = 0.001$)	0.55	30.15	0.95

Table 1: A comparison of the mean inversion scores of several ranking algorithms.

- S_i for the baseline and reply models outperforms all other ranking algorithms for all error measures.
- S_r for the baseline and reply models performs comparably to the indegree measure for I and I_N, and slighty outperform indegree for I_R.
- PageRank and the HITS authority score are outperformed by a significant margin. (The HITS hub score, not shown here, did rather worse than either.)

4.3 Sensitivity analysis

We evaluate the sensitivity of the rank orderings of models in this framework to the choice of input parameter values by building models using different parameter values, and then measuring the differences in rank orderings between each pair of models. We define the difference between two rank orderings as the mean absolute difference in rank ordering between each pair of individuals:

$$d(A_j, A_k) = \frac{\sum_{c \in C} |O_{A_j}(c) - O_{A_k}(c)|}{|C|} \quad (8)$$

where A_j and A_k are two different models, and $O_{A_j}(c)$ denotes the index of the rank assigned to c.

For a given entity set C, we observe that $d(A_j, A_k)$ takes on its maximum value when A_j produces an ordering that is the reverse of A_k's:

$$d_{\max}(A_j, A_k) = \frac{\sum_{k=1}^{|C|/2} (2k-1)}{|C|} = \frac{|C|}{2} \quad (9)$$

For this data set, therefore, the maximum mean difference is $628/2 = 314$; the figures below should be interpreted in the light of this information.

Figure 7 shows the result of cross-comparison of three ranking algorithms based on the basic model (S_i, S_o, and S_r), over variations in f. While varying f clearly has an effect on ranking–larger differences in f yield larger mean differences between the corresponding algorithms–the effect is small: the largest difference is $\approx$ 2% (6.02/314) for any pair of algorithms.

Figure 8 shows the result of cross-comparison of three ranking algorithms based on the reply model (S_i, S_o, and S_r), over variations in the parameters G and H, which are defined here to be functions of a single variable x: $G = 1800x$, and $H = 43200x$; larger values of x suggest an atmosphere in which email is generated and replied to at a slower pace. These figures indicate that varying x has a significant effect on rank ordering (again, larger differences in x yield larger mean differences in ordering): the largest mean difference in this case is $\approx$ 20% (65/314).

x	**0.5**	**1**	**2**	**4**
0.5	0.00	6.16	11.02	25.70
1	6.16	0.00	6.88	21.15
2	11.02	6.88	0.00	16.11
4	25.70	21.15	16.11	0.00

Table 2: Comparison of reply model algorithms ($f = 0.001$, varying x), for ranking method S_r.

The results shown in Figure 8 were generated using $f = 0.1$. We tested $f = 0.001$ as well, and found that the results for S_i and S_o (not shown here) were essentially identical, but that the results for S_r were markedly different; these results are shown in Table 2. We observe that a 100-fold reduction in f results in an approximately 2-fold reduction in mean difference magnitude for all values of x tested.

5. CONCLUSION AND FUTURE WORK

We have presented EventRank: a new framework, based on a set of clear requirements, for models that rank individuals in a social network derived from events occurring over time; these models respect event sequence and also provide a way of tracking rank changes over time as new events occur. Our experiments employed a novel method for evaluating the fitness of ranking algorithms when applied to a community with a known hierarchy, which involved evaluating the consistency of the rank ordering with the partial ordering specified by the organizational hierarchy.

Our preliminary investigation of this network, applied to an organizational email data set, has yielded promising results: our algorithms performed at least as well as the existing algorithms to which we compared them, and the orderings were shown to be a better fit with the organizational hierarchy.

Directions for future work include the following:

- application of these models to additional data sets, for further validation
- extension of the framework to incorporate header and content data; the reply model could be made more sophisticated, for example, if we knew which messages were in fact replies [7]
- application of this model to other types of event data, including undirected relations; the existing model does not depend on the fact that email events are directed
- investigation of methods for determining good values for f, G, and H based on requirements and time scales of interest
- analysis of transient rank values to automatically discover patterns in relative ranks of individuals over time, e.g., upward and downward trends in ranks for specific

f	0.9	0.1	0.01	0.001
0.9	0.00	4.44	5.79	6.02
0.1	4.44	0.00	1.81	2.12
0.01	5.79	1.81	0.00	0.44
0.001	6.02	2.12	0.44	0.00

f	0.9	0.1	0.01	0.001
0.9	0.00	3.72	4.69	4.87
0.1	3.72	0.00	1.30	1.64
0.01	4.69	1.30	0.00	0.50
0.001	4.87	1.64	0.50	0.00

f	0.9	0.1	0.01	0.001
0.9	0.00	2.18	3.12	3.37
0.1	2.18	0.00	1.26	1.58
0.01	3.12	1.26	0.00	0.42
0.001	3.37	1.58	0.42	0.00

Figure 7: Comparison of baseline model algorithms, varying f: S_i, S_o, and S_r respectively.

x	0.5	1	2	4
0.5	0.00	6.27	10.32	20.69
1	6.27	0.00	5.54	16.05
2	10.32	5.54	0.00	12.08
4	20.69	16.05	12.08	0.00

x	0.5	1	2	4
0.5	0.00	8.54	15.38	65.82
1	8.54	0.00	12.45	60.01
2	15.38	12.45	0.00	55.92
4	65.82	60.01	55.92	0.00

x	0.5	1	2	4
0.5	0.00	12.55	25.52	42.08
1	12.55	0.00	14.17	31.66
2	25.52	14.17	0.00	20.37
4	42.08	31.66	20.37	0.00

Figure 8: Comparison of reply model algorithms, $f = 0.1$, varying x: S_i, S_o, and S_r respectively.

individuals, periodic burstyness in ranks for individuals at certain times of year, etc.

6. ACKNOWLEDGMENTS

The authors wish to thank Danyel Fisher, as well as the reviewers, for their comments and feedback. This material is based upon work that was supported in part by the National Science Foundation under the Knowledge Discovery and Dissemination (KD-D) Program under Grant No. IIS-0083489.

7. REFERENCES

[1] S. P. Borgatti. Centrality and network flow. *Social Networks*, 27:55–71, 2005.

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

[3] 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.

[4] D. R. Gibson. Taking turns and talking ties: Networks and conversational interaction. *American Journal of Sociology*, May 2005. preprint.

[5] J. Heer. Exploring Enron: Visualizing ANLP results. *Applied Natural Language Processing*, 2004.

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

[7] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *IJCAI*, 2005.

[8] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y.-B. Boey. Analysis and visualization of network data using JUNG. http://jung.sourceforge.net/doc/JUNG_journal.pdf.

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

[10] The Mathworks. MATLAB, 2005.

[11] J. R. Tyler and J. C. Tang. When can I expect an email response? A study of rhythms in email usage. In *Proceedings of the Eighth European Conference on Computer Supported Cooperative Work*, pages 239–258, September 2003.

[12] F. Wu and B. Huberman. Discovering communities in linear time: A physics approach. *Eur. Phys. Journal*, B38:331–338, 2004.

[13] P. S. Yu, X. Li, and B. Liu. On the temporal dimension of search. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 448–449, New York, NY, USA, 2004. ACM Press.