Modeling Relational Events via Latent Classes

Christopher DuBois
Department of Statistics
University of California, Irvine
Irvine, CA 92697
duboisc@ics.uci.edu

Padhraic Smyth
Department of Computer Science
University of California, Irvine
Irvine, CA 92697
smyth@ics.uci.edu

ABSTRACT

Many social networks can be characterized by a sequence of dyadic interactions between individuals. Techniques for analyzing such events are of increasing interest. In this paper, we describe a generative model for dyadic events, where each event arises from one of C latent classes, and the properties of the event (sender, recipient, and type) are chosen from distributions over these entities conditioned on the chosen class. We present two algorithms for inference in this model: an expectation-maximization algorithm as well as a Markov chain Monte Carlo procedure based on collapsed Gibbs sampling. To analyze the model's predictive accuracy, the algorithms are applied to multiple real-world data sets involving email communication, international political events, and animal behavior data.

Categories and Subject Descriptors

I.5.1 [**Computing Methodologies**]: Pattern Recognition—*Statistical Models*

General Terms

Relational data, collapsed Gibbs sampling

1. INTRODUCTION

Social network analysis is the study of interactions among sets of entities, e.g. people, organizations, or nations. The dominant traditional approach to the statistical modeling of social networks focuses on graph-based representations with edges that persist indefinitely, such as friendships between individuals [10], or models where edges can be born and have long and indefinite durations, such as co-author relations from publication data [6] or online social networks [18].

In contrast, we focus on social network data that can be viewed as a set of *relational events* [5, 4]. Each event is an instantaneous or finite-duration action involving two or more entities. For example, one might model instant messaging data as a set of relational events, where each event is an instantaneous directed edge with a sender node and a receiver node. This type of relational data has received far less attention in the data mining and social network literature than static network data. Nonetheless dynamic network data is becoming increasingly common, particularly in a world with many different digital modes of time-stamped communications, e.g., Facebook comments, email messages, and instant messaging.

In particular, in this paper, we investigate the problem of predicting the rate at which individuals will send and receive events in the future, given historical event data. We focus on dyadic instantaneous events. We will assume a stationary Poisson process for event generation—the extension to non-stationarity is likely to be important for practical applications but is beyond the scope of the present paper. For each event let $s \in \mathcal{S}$ denote the identity of the sender, $r \in \mathcal{R}$ denote the identify of the receiver, and $a \in \mathcal{A}$ denote the type of the event. We begin by initially focusing on the common situation where the sets of possible senders and receivers are the same ($\mathcal{S} = \mathcal{R}$ and $|\mathcal{S}| = n$) and there exists a single type of event ($|\mathcal{A}| = 1$)—we will return to the more general situation later in the paper.

Let λ_{sr} be the (unknown) Poisson rate of event generation between sender s and receiver r. Assuming independence of the pairwise Poisson processes, the superposition of independent Poisson processes is itself Poisson with rate $\lambda = \sum_s \sum_r \lambda_{sr}$: this is the rate at which events are generated in the network as a whole. One can show the probability that the next event in the network corresponds to edge (s, r) can be written as

$$P(s,r) = \frac{\lambda_{s,r}}{\sum_{s'} \sum_{r'} \lambda_{s',r'}} = \frac{\lambda_{s,r}}{\lambda}$$

i.e., the probability of the edge (s, r) occurring next is equal to the Poisson rate of (s, r) divided by the total rate for the network [21]. The set of $P(s, r)$ probabilities sums to 1. Thus, one representation of the network process is that (a) events occur globally in the network with rate λ, and (b) the probability that each such event involves a specific pair s, r is determined by a multinomial distribution over n^2 possible pairs.

We focus on the problem of modeling $P(s, r)$ for event data (and more generally $P(s, r, a)$ where a indicates event type). Note that we can convert these probabilities to event rates $\lambda_{s,r}$ simply by multiplying our estimate of $P(s, r)$ by the network rate λ. For large networks with many events, estimating the global rate of event generation λ will be relatively easy. In contrast, estimating $P(s, r)$ will generally be much more difficult since there are n^2 such event-pairs.

A simple baseline approach to estimating the $P(s, r)$ matrix is to simply count the number of entries observed for each pair s, r in the historical data set and then use a frequency-based estimate for

KDD'10, July 25–28, 2010, Washington, DC, USA.

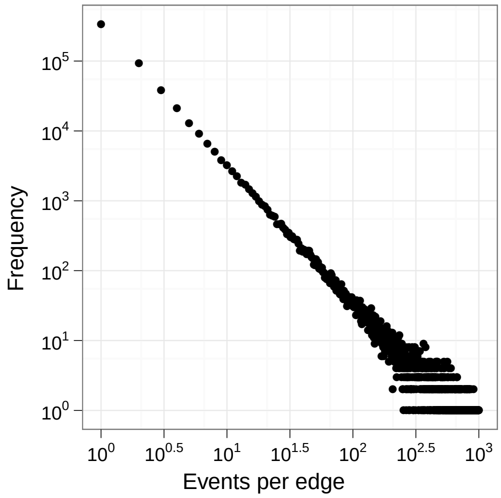


Figure 1: Number of edges for which we observe a given number of events in a data set of international dyadic events [15].

prediction, or a smoothed version of the same, e.g.,

$$\hat{P}(s,r) = \frac{N_{s,r} + \alpha}{\sum_{s'} \sum_{r'} (N_{s',r'} + \alpha)}, \quad 1 \leq s,r \leq n \tag{1}$$

where $N_{s,r}$ is the number of observed events from s to r in the training data, and α is a smoothing parameter (e.g. from a symmetric Dirichlet prior). A fundamental problem in this context is sparsity: many pairs s, r will have observed counts of $N_{s,r} = 0$ even though there may be a non-zero probability of actor s sending an event to actor r. For example, in a particular university, researcher A and researcher B might never have communicated in the past, but they may well communicate in the future. The problem is particularly acute in the so-called "cold-start" scenario, where we are observing a new network (e.g., a new class of students at a university) when we have very small amounts of sparse event data on which to base our predictions. The use of smoothing parameters in the probability estimates (e.g., the use of α above) will certainly ameliorate the situation, but these smoothing parameters will tend to spread the probability mass evenly over all possible events. This will not be helpful for prediction in large sparse networks where the probability of events between pairs is highly skewed, i.e., some pairs of individuals will have a much higher probability of communicating, as seen in Figure 1.

A natural idea in this context is to learn groupings of the individuals and "borrow strength" from relevant groups when estimating individual pair probabilities. For example, say graduate student A is in group 1 and graduate student B is in group 2. If there is evidence that groups 1 and 2 collaborate then we might predict that A and B will communicate in the future, even if they have not done so in the past. A well known approach of this type in social network analysis is *stochastic blockmodeling*. A *blockmodel* is defined as a mapping of approximately equivalent actors into blocks, along with a statement regarding the relations between the blocks [2]. Stochastic blockmodels have a rich history in the statistical modeling of social networks [8, 13, 26], both for exploratory analysis and answering substantive questions. Statistical learning approaches can be used to infer both likely partitions of actors and the probability of block-wise interactions [25].

However, stochastic blockmodels are typically used to model static binary relationships among individuals, e.g., binary edges indicating friendship. In contrast, we are interested in the relative frequency of events over time between pairs of individuals. Specifically we model pairwise probabilities $P(s,r)$, namely the probability that the next event in the network will occur between s and r, rather than modeling the probability that an edge exists in a static context between s and r, e.g., $P(e_{s,r} = 1)$.

The specific contributions of this paper can be summarized as follows. We propose a latent class model that is similar in spirit to stochastic blockmodels but that is designed to model relational event data. We demonstrate the capabilities of the model on both simulated and real data sets, interpreting the latent class information that the model extracts from data. Furthermore, we show how one can directly assess the predictive accuracy of these models and illustrate how our proposed approach can be readily extended to make inferences in the presence of missing event information, e.g., learning from events where we know the sender of an event but do not know the receiver.

The organization of this paper is as follows. After presenting the model formally, we discuss two methods for performing inference. A simple illustration of the model's use on simulated data is presented, followed by an evaluation of its predictive performance on multiple real world data sets. We conclude with a discussion of the empirical results and future directions.

2. THE MARGINAL PRODUCT MIXTURE MODEL

Motivated by dyadic interaction, we propose a model for events with a sender, receiver, and action type. Formally, we consider a possible set of events $E = \{(s,r,a) : s \in \mathcal{S}, r \in \mathcal{R}, a \in \mathcal{A}\}$ where $\mathcal{S}, \mathcal{R}, \mathcal{A}$ are the sets of possible senders, receivers, and action types respectively. The observed data then form a sequence of T events, which we denote $\mathcal{D} = \{e_i : e_i \in E, i = 1, \ldots, T\}$.

As discussed earlier, our goal is to estimate the probabilities $P(s,r,a)$ that the next event in the network will be from sender s to receiver r and of type a, i.e., a multinomial consisting of $n_s \times n_r \times n_a$ probabilities that sum to 1. Rather than modeling individual triples (s,r,a), the approach we take is to hypothesize the existence of a finite set of latent classes (or clusters) for events, each characterized by conditionally-independent marginal distributions over senders $\mathcal{S}$, receivers $\mathcal{R}$, and actions $\mathcal{A}$. This allows us to approximate the full array $P(s,r,a)$ with a parsimonious mixture of simpler distributions that require far fewer parameters to specify than the unconstrained model with $O(n_s \times n_r \times n_a)$ parameters.

We assume that events are exchangeable within a *latent class* of events. We further assume the event's sender, receiver, and action type are conditionally independent given the latent class. Under this model, each new edge arises from latent class c with probability π_c; next, the attributes of the edge are drawn from associated multinomial distributions over likely senders, receivers, and action types for the given class (where θ, ϕ, ψ respectively parameterize each of these distributions). For example, $\phi_{c,s} = P(s|c)$ is the probability of selecting sender s given class c. We use standard non-informative Dirichlet priors on these multinomials, allowing for straightforward derivation of posterior distributions of interest.

This results in the following simple generative model for sets of relational events with C latent classes, where $\vec{\alpha}$, $\vec{\beta}$, $\vec{\gamma}$, and $\vec{\delta}$ are the parameters of associated Dirichlet priors:

1. Draw the class distribution $\vec{\pi} \sim$ Dirichlet($\vec{\alpha}$)

2. Draw distributions: $\vec{\theta}_c \sim$ Dirichlet($\vec{\beta}$), $\vec{\phi}_c \sim$ Dirichlet($\vec{\gamma}$), $\vec{\psi}_c \sim$ Dirichlet($\vec{\delta}$) for all $c \in \{1, \ldots, C\}$

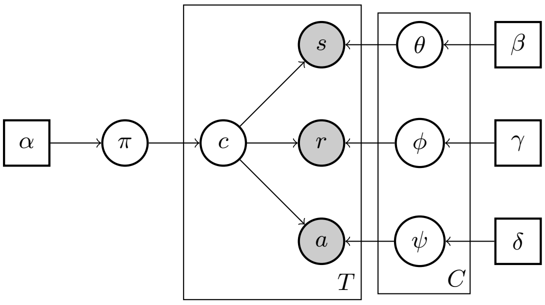


Figure 2: Graphical model for the MPMM.

3. For each event
   (a) Draw $c \sim$ Multinomial($\vec{\pi}$), the event's class
   (b) Draw $s|c \sim$ Multinomial($\vec{\theta}_c$), the event's sender
   (c) Draw $r|c \sim$ Multinomial($\vec{\phi}_c$), the event's receiver
   (d) Draw $a|c \sim$ Multinomial($\vec{\psi}_c$), the event's type

The graphical model is shown in Figure 2. Note that the parameter vector $\vec{\pi}$ has dimension C, and for each c the vectors $\vec{\theta}_c$, $\vec{\phi}_c$, and $\vec{\psi}_c$ have dimension n_s, n_r, and n_a respectively. From the above generative model we can immediately derive the likelihood, where $\mathbf{\Phi}$ is the set of all parameters in the model:

$$\begin{aligned} P(\mathcal{D} \mid \mathbf{\Phi}) &= \prod_{t=1}^{T} \sum_{c=1}^{C} P(e_i = (s_i, r_i, a_i), c_i = c | \mathbf{\Phi}) \\ &= \prod_{t=1}^{T} \sum_{c=1}^{C} P(s_i | \vec{\theta}_c) P(r_i | \vec{\phi}_c) P(a_i | \vec{\psi}_c) P(c | \vec{\pi}) \\ &= \prod_{t=1}^{T} \sum_{c=1}^{C} \theta_{c,s_i} \phi_{c,r_i} \psi_{c,a_i} \pi_c \qquad (2) \end{aligned}$$

One may interpret the above expression as follows: each event's probability incorporates the product of the probabilities of its sender, receiver, and action type given its latent class; we then sum over the possible latent classes, weighting by the probability of each class. Note that for each latent class c the model predicts the probability of edge (s, r, a) as the product of marginal distributions. Thus the model can be conceptualized as a marginal product mixture model (MPMM) and is related to recent work on factorized representations for multi-view data sets [20].

It is informative to look at the representational capabilities of this model compared to traditional blockmodels as used in social network modeling. For example, consider the sociomatrix in Figure 3a, where each element (s, r) denotes the probability of the next event being sent by actor s and received by actor r. This pattern of interactions could be described by a blockmodel (as shown in Figure 3c) in terms of a partition of the senders, a partition of the receivers, and block-wise probabilities. This blockmodel would require $6 \times 7 = 42$ parameters to capture these blockwise interactions. In our model, however, we see a more parsimonious explanation in terms of *classes* of activity. As shown in Figure 3b, there are four classes of events where events within each class occur with the same probability.

The model is general enough to handle several useful special cases. For instance, the model allows for asymmetric behaviors among nodes; a given individual might initiate more events under one class and receive more events under another class. When there is a single event type, then $n_a = 1$: step 3d of the generative model is ignored and the derivations and algorithms require trivial modification. When events are undirected, we use a single set of parameters for both senders and receivers. Also, the situation $|\mathcal{S} \cap \mathcal{R}| = \emptyset$ (ie. a bipartite graph) requires no modifications to the model.

While the MPMM makes strong assumptions regarding the conditional independence of edge attributes and does not incorporate any sequential dependence, we show such methods can be useful for exploring and modeling large, real-world data sets.

3. INFERENCE

We wish to infer the parameters of the model and the latent class assignments for events, given observed data $\mathcal{D}$ and the likelihood of Equation 2. This is a typical mixture model likelihood for which there is no closed-form for the posterior distribution and therefore we must resort to approximate inference methods. We present two algorithms for learning the posterior distribution of the latent class assignments as well as point estimates of the parameters Φ: a collapsed Gibbs sampler (CGS) and an expectation-maximization (EM) algorithm.

3.1 Collapsed Gibbs Sampling

First we provide a Markov chain Monte Carlo algorithm that uses Gibbs sampling to iteratively simulate the conditional posterior distribution of the latent classes. For each event, we need the distribution over possible classes conditioned on everything else. Note that since the model uses conjugate priors we can integrate out $\vec{\pi}$, $\vec{\theta}_c$, $\vec{\phi}_c$, and $\vec{\psi}_c$ for all c in closed form (and avoid the uncertainty associated with them while sampling). We are left with the following conditional distribution which may be used to sample the latent assignment for observation i:

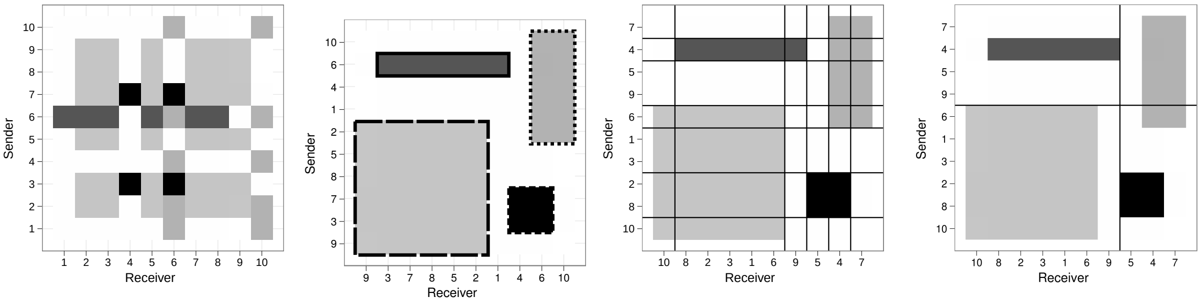
$$\begin{aligned} P(c_i = c | c^{\neg i}, \mathcal{D}, \Phi) \quad &\propto \left(M_c^{\neg i} + \alpha\right) \left(\frac{U^{\neg i}_{c,s_i} + \beta}{\sum_{s=1}^{n_s} U^{\neg i}_{c,s} + n_s \beta}\right) \\ &\left(\frac{V^{\neg i}_{c,r_i} + \gamma}{\sum_{r=1}^{n_r} V^{\neg i}_{c,r} + n_r \gamma}\right) \left(\frac{W^{\neg i}_{c,a_i} + \delta}{\sum_{a=1}^{n_a} W^{\neg i}_{c,a} + n_a \delta}\right) \end{aligned}$$

where $M_c = \sum_i I(c_i = c)$, $U_{c,s} = \sum_i I(c_i = c, s_i = s)$, $V_{c,r} = \sum_i I(c_i = c, r_i = r)$, and $W_{c,a} = \sum_i I(c_i = c, a_i = a)$. The parameters α, β, γ, and δ are smoothing parameters from symmetric Dirichlet priors. Note $c^{\neg i} \equiv \{c_j : j \neq i\}$, so $M_c^{\neg i} = \sum_{j \neq i} I(c_j = c)$ for example. The derivation of the above sampling equation follows closely to that of collapsed Gibbs sampling for latent Dirichlet allocation [11]. Given the class assignments for all events, we can compute estimates for the model parameters.

$$\begin{aligned} \hat{\pi}_c &= \frac{M_c + \alpha}{\sum_c M_c + C\alpha} \\ \hat{\theta}_{c,s} &= \frac{U_{c,s} + \beta}{\sum_{s=1}^{n_s} U_{c,s} + n_s \beta} \\ \hat{\phi}_{c,r} &= \frac{V_{c,r} + \gamma}{\sum_{r=1}^{n_r} V_{c,r} + n_r \gamma} \\ \hat{\psi}_{c,a} &= \frac{W_{c,a} + \delta}{\sum_{a=1}^{n_a} W_{c,a} + n_a \delta} \end{aligned}$$

Algorithm 1 shows the procedure in full. We assess convergence by monitoring the log-likelihood of the training data under the model. While 20-30 iterations often appears sufficient for convergence, in all the experiments that follow we use 1000 iterations.

We obtain better predictive performance by running multiple chains of the CGS algorithm and averaging over the posterior predictive distributions for the next edge (we will refer to this as MCGS).



(a) (b) (c) (d)

Figure 3: (a): A matrix of observed counts for all possible events where darker shades indicate higher frequency. (b): MPMM models the data via a low-dimensional representation indicating groups of events. This structure is apparent after reordering the rows and columns. Stochastic blockmodels can model such data either by having (c) many small homogeneous blocks or (d) with a few inhomogeneous blocks; the latter misses some of the structure.

Algorithm 1 CGS Algorithm for MPMM

```
for $i = 1$ to $T$ do
  Initialize $c_i$ with random integer between 1 and $C$
end for
Compute count matrices $M, U, V, W$
while not converged do
  for $i = 1$ to $T$ do
    Decrement $M[c_i], U[c_i, s_i], V[c_i, r_i], W[c_i, a_i]$
    for $c = 1$ to $C$ do
      $\eta[c] \leftarrow (M[c] + \alpha)\left(\frac{U[c,s_i]+\beta}{\sum_c U[c,s_i]+n_s\beta}\right)$
        $\left(\frac{V[c,r_i]+\gamma}{\sum_c V[c,r_i]+n_r\gamma}\right)\left(\frac{W[c,a_i]+\delta}{\sum_c W[c,a_i]+n_a\delta}\right)$
    end for
    $c_i \leftarrow$ IndexOf(RandomMultinomial$(1, \eta)$)
    Increment $M[c_i], U[c_i, s_i], V[c_i, r_i], W[c_i, a_i]$
  end for
end while
$\hat{\pi} \leftarrow$ Normalize$(M + \alpha)$
$\hat{\theta} \leftarrow$ NormalizeColumns$(U + \beta)$
$\hat{\phi} \leftarrow$ NormalizeColumns$(V + \gamma)$
$\hat{\psi} \leftarrow$ NormalizeColumns$(W + \delta)$
```

3.2 Expectation-Maximization Algorithm

As an alternative to CGS, we derive an EM algorithm that provides the marginal posterior density for the latent class assignment for each event. The algorithm iteratively maximizes the expected complete-data loglikelihood (which includes the latent class information) thus giving us the maximum likelihood estimates for the parameters. After randomly initializing the $P(c_i = c)$ we iteratively perform the following computations for all $c \in \{1, \ldots, C\}$:

E-step:

$$P(c_i = c|s_i, r_i, a_i, \Phi) \propto \theta_{c,s_i}\phi_{c,r_i}\psi_{c,a_i}$$

M-step:

$$\hat{\theta}_{c,s} = \frac{\sum_{i=1}^{T} I(s_i = s)P(c_i = c) + \beta}{\sum_{i=1}^{T} P(c_i = c) + n_s\beta}$$

$$\hat{\phi}_{c,r} = \frac{\sum_{i=1}^{T} I(r_i = r)P(c_i = c) + \gamma}{\sum_{i=1}^{T} P(c_i = c) + n_r\gamma}$$

$$\hat{\psi}_{c,a} = \frac{\sum_{i=1}^{T} I(a_i = a)P(c_i = c) + \delta}{\sum_{i=1}^{T} P(c_i = c) + n_a\delta}$$

We continue iterating until the log-likelihood between iterations changes by less than $\epsilon = 1e - 8$ for the experiments in this paper, and repeat this 5 times.

3.3 Scalability

Space and time complexity of the learning algorithms is important for large data sets. Inference and prediction for the MPMM is well-suited to sparse data since the likelihood is only defined over events that occurred, rather than over all events that could have occurred. Because of this, the training time for our proposed approach scales linearly in the number of observed events T. In contrast, other statistical network models (such as stochastic blockmodels [19] and latent-space models [12]) scale as $O(n^2)$ or worse, since the likelihood is defined over all pairs of individuals, whether an edge between them exists or not. For large sparse networks, scaling as $O(T)$ is likely to be much more computationally efficient than scaling as $O(n^2)$. For CGS, the space complexity is $O(T + C(n_s + n_r + n_a))$ and the time complexity is $O(TC)$ per iteration as each Gibbs scan requires only a single pass through the data set. The time complexity for the EM algorithm is $O(TC(n_s+n_r+n_a))$ per iteration and the space complexity is $O(TC+C(n_s+n_r+n_a))$.

4. AN ILLUSTRATIVE EXAMPLE

Finding latent event classes may be a helpful tool for exploratory data analysis in large data sets, in a manner similar to how topic models can facilitate the grouping of collections of documents by latent topics [11].

To illustrate the use of our model for exploratory data analysis, we use a data set of international events involving entities from 450 countries over the 2000-2005 time period [15]. This data has been used by political scientists to explore international relations and policy. The authors used an automated system for coding 3,575,897 events from Reuters news reports. Each of these events takes the form: *[entity A] [action] [entity B]*. Actions in this data set consist of 247 possible types, such as judicial action, military action, and so forth. This data is well suited to the MPMM approach; not only does it contain a very large number of possible edges (roughly $13,000 \times 13,000 \times 247$) which is difficult to model using standard social network analysis methods, but it also distinguishes between different types of activity.

To illustrate how the MPMM can work on bipartite graphs, we consider a subset of the data concerning US international relations, restricting senders to be US-based and recipients to be foreign. After applying MCGS with $C = 50$ (see Section 5.1 for a discussion of hyperparameter settings), we can explore typical senders, receivers, and action types as shown in Table 1 for three particular classes.

Top Senders	Pr.	Top Receivers	Pr.	Top Actions	Pr.
Class A					
United States : Government agents	0.47	Greece : NA	0.05	Sports contest	0.59
United States : Athletes	0.29	Australia : Government agents	0.02	Agree or accept	0.14
United States : Nominal agents	0.04	United Kingdom : NA	0.02	Optimistic comment	0.04
United States : Police	0.04	Canada : Government agents	0.02	Comment	0.03
United States : Occupations	0.04	France : NA	0.01	Control crowds	0.03
United States : Ethnic agents	0.03	Belgium : Government agents	0.01	Improve relations	0.01
Class B					
United States : Military	0.88	Iraq : Government agents	0.17	Comment	0.19
United States : Government agents	0.08	Iraq : National executive	0.07	Military raid	0.14
United States : Military hardware	0.01	Iraq : Military	0.05	Military clash	0.10
United States : Officials	0.00	Iraq : Ethnic agents	0.05	Military occupation	0.10
United States : Police	0.00	Iraq : Intangible things	0.04	Shooting	0.10
United States : Motor vehicles	0.00	NA : Insurgents	0.04	Political arrests and detentions	0.04
Class C					
Top Senders	Pr.	Top Receivers	Pr.	Top Actions	Pr.
United States : National executive	0.73	Palestine : National executive	0.22	Discussions	0.44
United States : Diplomats	0.15	Israel : National executive	0.12	NA	0.22
United States : Government agents	0.06	Israel : Government agents	0.09	Call for action	0.09
United States : Human actions	0.01	Egypt : National executive	0.06	Demand	0.04
United States : Artists	0.01	Palestine : Government agents	0.04	Collaborate	0.03
United States : Occupations	0.01	India : Government agents	0.03	Host a meeting	0.03

Table 1: Excerpts from the sender, receiver, and action type distributions for latent classes of international dyadic events as learned by the MPMM.

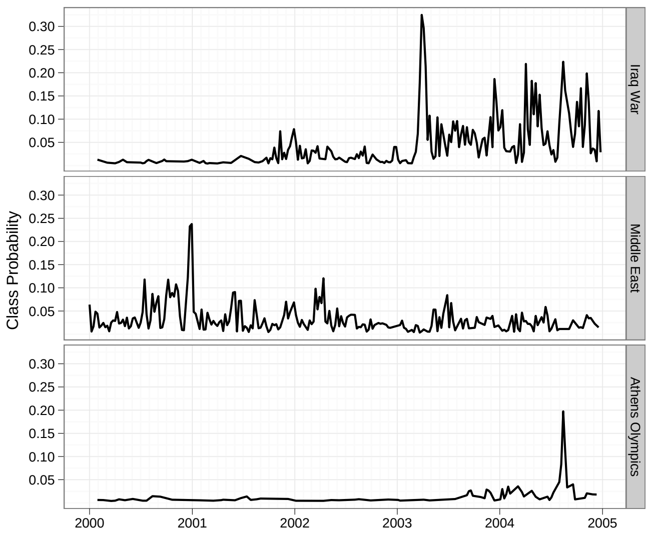


Figure 4: Proportion of assignments per week for latent classes from a data set of international political events [15], with typical senders, receivers, and action types shown in Table 1.

The first is a class of events mostly concerning the 2004 Olympics, the second concerns the Iraq war, and the third primarily concerns the Middle East conflict. Action types are also clustered simultaneously; the action types with the MPMM upon inspection the action types seem appropriate given each class (e.g. "Sports contest" actions should grouped in a class that often includes "Athletes"). The "Comment" action type encompasses information from interviews or public statements.

Although the MPMM uses only sets of counts (rather than sequential or temporal information) we can nonetheless retroactively examine the timeline of class assignments for each event after we have fit the model. In Figure 4 we plot the number of events per week assigned to each of the three latent classes from Table 1. Note that spikes in the plot have correspondence with known world events, such as the beginning of the Iraq war early in 2003 and the Olympic Games in 2004.

5. EXPERIMENTAL METHODS

In this section we empirically study the MPMM's predictive ability, using baseline models and different data sets.

5.1 Algorithms and Settings

We consider several models for comparison. A trivial baseline approach for relational event data is to predict all possible events with equal probability. We will refer to this baseline as `Uniform`.

Another simple approach is to make predictions using the observed frequency of each event. For a data set of T events, one can view the aggregated counts, $\vec{Y}$, as a Multinomial(T,$\vec{p}$) random variable, where $\vec{p}$ has length $n_r n_s n_a$. We denote this method `Multinomial`.

The maximum likelihood estimate (MLE) for $\vec{p}$ is $\vec{Y}/T$. This estimate is asymptotically unbiased but it will suffer from high variance when applied to finite sparse data sets since many cells will have no data. By placing a Dirichlet prior on $\vec{p}$ we can smooth our probability estimates over unobserved events. In practice we let $\vec{p} \sim$ Dirichlet(η) with $\eta = \frac{Q}{n_s n_r n_a}$ where Q determines the prior's effective sample size. In our experiments we set $Q = 100$ to keep the effect of the prior consistent across experiments. To place the MPMM on equal footing with this baseline, we set the MPMM hyperparameters accordingly, letting $\alpha = 2$ and $\beta = \gamma = \delta = \eta^{1/3}$. It is straightforward to show that, with these hyperparameters, the priors for both `Multinomial` and MPMM assign the same probability to a particular edge (s, r, a).

Other models, such as the infinite relational model (IRM) [14], often assume entities belong to one or more latent clusters. Such models can be adapted to the type of event data considered in this paper by modeling the probability of each combination of clusters. In our experiments we compare our model to the IRM approach by using a single relation on three domains. Specifically, we model $Y_{s,r,a} = \frac{W_{k_s,k_r,k_a}}{|k_s||k_r||k_a|}$ where k_s is the cluster assigned to s by the IRM, $|k_s|$ is the size of the cluster, and $W_{k_s,k_r,k_a}|\theta \sim$ Multinomial(T, θ), where $\theta \sim$ Dirichlet(α). Following [14] we let $\alpha = \beta|k_s||k_r||k_a|$ where $|k_s|$ is the size of cluster k_s and $\beta = .1$.

5.2 Prediction

Since we have defined a probabilistic model for how events are generated, we can compute the predictive probability of a future event using parameters estimated from training data. For example, suppose we want to know the probability of a particular event (s, r, a). Substituting model parameter estimates into Equation 2, we can compute

$$\hat{p}_{s,r,a} = P(s, r, a|\mathcal{D}, \hat{\pi}, \hat{\theta}, \hat{\phi}, \hat{\psi}) = \sum_c \hat{\pi}_c \hat{\theta}_{c,s} \hat{\phi}_{c,r} \hat{\psi}_{c,a}$$

In the case of MCGS, each $\hat{p}_{s_i,r_i,a_i}$ is computed by averaging Z estimates of $\hat{p}$, taking the last sample obtained from Z independent CGS chains, where $Z = 20$ in the experiments below.

5.3 Evaluation

To evaluate the predictive performance of our model, we compute the average log probability of observed events in a heldout test set of T observations (e.g., for general motivation see [9]):

$$L_{\text{test}} = \frac{1}{T}\sum_{i=1}^{T} \log(f(Y_i|Y_{train})) = \frac{1}{T}\sum_{i=1}^{T} \log(\hat{p}_{s_i,r_i,a_i})$$

If model A has a larger value of L_{test} compared to model B, this is evidence that model A is a better predictive model than model B, and is assigning higher probabilities to edges that actually occur in the test set and lower probabilities to those that do not occur (compared to model B). Alternative performance scores could also be used, although such scores (such as mean-squared error) often put an over-emphasis on events that did not actually occur (i.e., all pairwise "non-events" at each time-step).

6. EXPERIMENTAL RESULTS

The top row of Figure 5 is a visual representation of a particular MPMM which we used in simulation experiments. The 4 panels correspond to $n_a = 4$ different action types. The rows and columns in each panel represent $n_s = n_r = 100$ senders/receivers. Darker shades indicate a higher probability of that edge occurring. The shaded rectangles correspond to 4 different latent classes of events, A, B, C, D. For example, the third action type (third panel) is associated with latent classes B, C, and D.

We created synthetic training data sets of various sizes by simulating from the MPMM with these parameters, and compared the predictive performance of different models on independent test data. The lower-left panel in Figure 5 shows how the predictive performance (test log-likelihood) on the test data varied for each model as a function of training data size (on a log-scale). The different implementations of MPMM (EM, CGS, MCGS) were not significantly different from each other, so we only show MCGS for clarity. The upper line in the graph is the test log-likelihood for the model with the "perfect" true parameter values (no learning). The lower line is the performance of the Uniform model (which ignores the training data). Between these two extremes, we see that the MPMM dominates the Multinomial, as we would expect given that the data is being generated from an MPMM. The multinomial model initially has poorer predictive ability as the training set size goes from 10^2 to 10^3—this may be due to initial overfitting on small training data sets as the effect of the smoothing prior weakens. As it sees more data (beyond 10^3) it starts to improve and gradually catches up with the best-performing MPMM models. We used different values of C with MPMM to show its effect on predictive performance; since we generated the synthetic data using 4 latent classes, it is unsurprising that using $C = 3$ does not perform as well.

To gain some intuition for the MPMM, it is helpful to consider predictions made for a given node. Figure 5 shows the probability of a particular actor (number 30) initiating an edge to each of the possible receivers for the synthetic data set. The probabilities are quite different for each model: with the MPMM, we are effectively smoothing over the set of typical receivers for those edges where actor 30 takes part. The true distribution is included for comparison.

We additionally consider three different real-world data sets. The first consists of $T = 200,000$ records of emails from a European university collected over 83 days among $n_s = n_r = 2562$ individuals [7]. To create an action type, a, we discretized the log of the size of the email message (in kB) into $n_a = 10$ bins.

The second data set involves dominance acts among $n_s = n_r = 63$ red deer stags in Scotland [3]. Red deer engage in aggressive acts to enforce a social hierarchy. The data consists of $T = 1200$ observed aggressive actions (e.g. glaring, kicking, and mounting), where $n_a = 10$. Each event had a clear "winner" and "loser", which we code as the sender and recipient, respectively.

The third data set is the international events data discussed earlier in the paper. We evaluated the algorithms on the same subset of events considered in Section 4, comprising a total of $n_a = 81$ senders from the USA, $n_r = 2695$ non-US recipients, and $n_a = 178$ actions (types of events). The total number of events is $T = 40031$.

The left panel of Figure 6 shows the predictive performance of different models as a function of training set size for the 3 data sets: the red deer data set (with a test set of 200 observations), the email data set (with a test set of 100,000 observations), and the international political events data set (with a test set of 10,000 observations). We take the mean of the posterior predictive distribution over 20 chains of the CGS algorithm for MPMM using $C = 3$, 20, and 50. Hyperparameter selection is dicussed in Section 5.1. For each of 10 runs, the training and test sets are randomly sampled and we plot the mean test log-likelihood across runs (with the error bars showing 95% confidence intervals). With large amounts of training data the Multinomial baseline steadily improves as expected, but the MPMM has significantly better predictive power for a wide range of training set sizes. For the red deer data set, the IRM outperforms the Multinomial baseline, but does not perform as well as the MPMM.[1]

7. PREDICTION WITH MISSING DATA

We also measure the predictive performance of the MPMM when some of the events are only partially observed. Missing data is a well-studied issue in social network analysis, but most effort has been motivated by survey data, e.g. accounting for censoring, network boundaries, and so on [16]. Both CGS and EM can be extended in a straightforward manner to make inferences over missing sender, recipient, or type information. Here we just show results using EM as the inference technique.

The right-hand side of Figure 6 compares the predictive perfor-

[1] We were unable to obtain experimental results for the IRM on the two larger datasets (email and international political events).

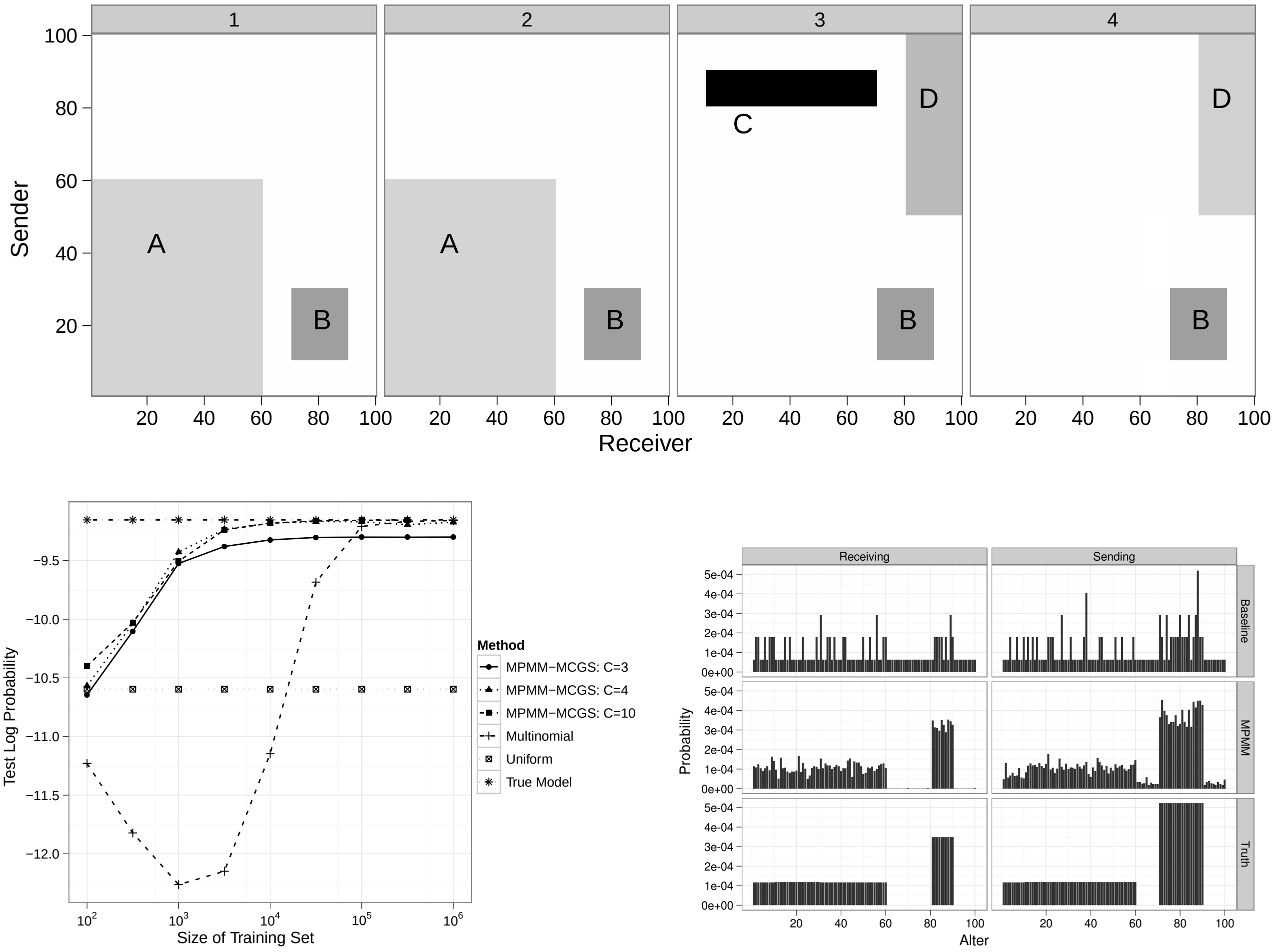


Figure 5: Top: Probability of an event for a synthetic data set with four latent classes labeled A, B, C, and D. Each panel represents an action type. Left: Comparison of test log-likelihood for the synthetic data set using the CGS algorithm with different numbers of latent classes C versus the baseline. See 6 for a discussion. Right: Comparison of predictive distributions between models and truth for actor 30 on a training set of 10000 events with the hyperparameters used in the experiment at left. The MPMM approximation provides a better fit over the typical recipients.

mance on test data of learning MPMMs with fully-observed data versus partially-observed data. For this experiment we split each of the real data sets into (a) a test set of the same size used in the previous experiments, and (b) training sets of various sizes. All models were given a fixed number of events that were fully observed (corresponding to the left most data point in the graphs on the right-hand side of Figure 6). The “Complete” model was then given additional observations (corresponding to increasing values along the x-axis). The “Incomplete” models were also given additional observations, again corresponding to the x-axis, but in this case the additional events had missing recipient IDs (i.e., events are only partially observed). All models were trained using EM with $C = 5$, where for partially observed data the recipient IDs were treated as missing and probability distributions over recipients for each event were estimated and maintained in a standard EM fashion.

The goal of the experiment was to determine if the MPMM approach could extract useful information from event data even when information such as recipient IDs were missing. The figures show that the test log-likelihood indeed increases as additional partially observed data is provided to the learning algorithm. The performance improvement is not as good as that obtained with fully-observed data (top line for each data set) but nonetheless is significantly better than models that ignore the partially observed data (the left-most points on the graphs). Note that probabilistic models are particularly useful in the context of missing information—for non-probabilistic modeling approaches it would be difficult to incorporate the partially observed data into a learning algorithm.

8. DISCUSSION AND RELATED WORK

The model proposed here is simple in that each event’s sender, receiver, and action type are assumed to be conditionally independent given the latent class for that event. However, in return for these strong assumptions, we get relatively straightforward and scalable inference algorithms, making the learning algorithms practical as a tool for analyzing large event data sets.

Extending the model to allow for time-dependence is a natural direction for future work. For example, it is straightforward to add Markov dependence for the latent classes, resulting in a hidden

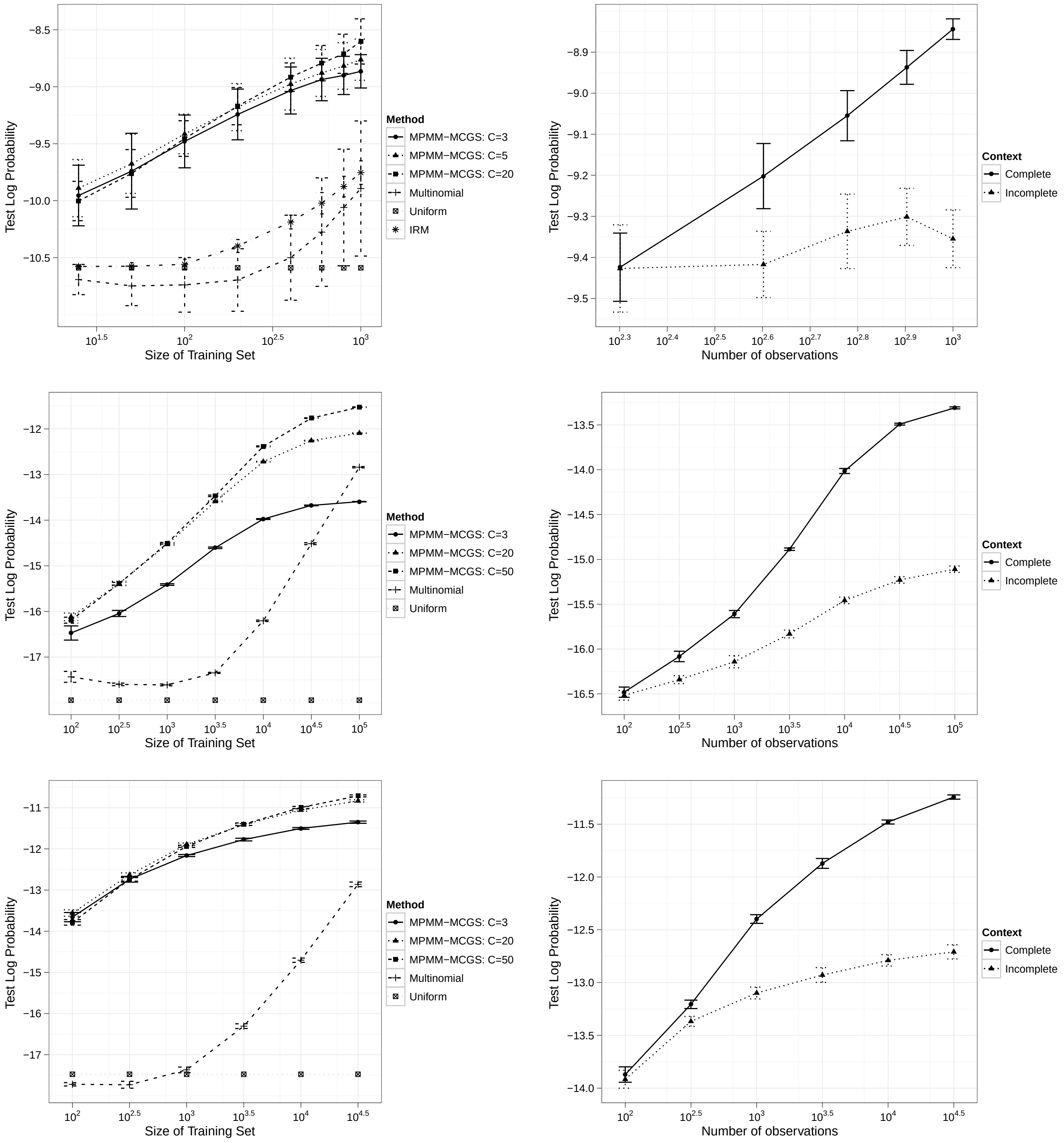


Figure 6: The left-hand plots show the predictive accuracy (measured by test log-likelihood) of different models on real-world data sets. The right panel compares the performance of MPMM with fully observed event data versus partially observed data. Top: Red Deer data set. Center: Email data. Bottom: International dyadic events.

Markov model—this kind of temporal dependence may be useful for networks where the entire network undergoes "global" changes in behavior (e.g., modeling patterns in team sports). Similarly, it would be straightforward to make the hidden process dependent on exogenous time-series representing external influences on the network, or dependent on time directly such as time of day, day of week, or time of year. Modeling dependence at the individual actor and event level is somewhat more challenging, and approaches such as network-based sufficient statistics may be useful [5].

Also of interest are extensions to models that can handle events with durations, e.g., the special case of events that are "born" and assumed to persist indefinitely, e.g. the growth of a friendship graph on a social networking site. One approach for modeling such phenomena is to combine models of user-centric activity (as in Leskovec et al. [18]) with stochastic models such as the mixed membership stochastic blockmodel [1] (see below) or MPMM to automatically share statistical strength among similar groups of nodes or edges.

The approach we present in this paper differs from much of the prior work on block modeling in that here we model latent classes of events, rather than latent classes of individuals. The Bayesian formulation of the stochastic blockmodel (first proposed by Holland, Laskey, and Leinhardt [13] and developed further in Snijders and Nowicki [25] and Nowicki and Snijders [19]), focuses on the assignment of nodes to latent classes as well as estimation of blockwise interaction rates. In that model, each node i is first assigned to a latent class: $z_i \sim \text{Multinomial}(1, \theta)$. Each possible edge is drawn using a $K \times K$ matrix, η, that describes the probability of the two blocks interacting: $Y_{ij} \sim \text{Bernoulli}(\vec{z_i}^T \eta \vec{z_j}), 1 \leq i, j \leq n$.

The Infinite Relational Model (IRM) of Kemp et al. [14] extends the Nowicki and Snijders approach [19] to networks with edges having more than two nodes (i.e., not just sender and receiver). Kemp et al. use a Chinese Restaurant Process (CRP) for assigning each node to a latent group; this nonparametric method allows the number of latent classes to be flexible and data-dependent. Kurihara, Kameya, and Sato [17] further extended the IRM to model events by picking the sender and receiver from a multinomial distribution over possible nodes.

The Mixed Membership Stochastic Blockmodel (MMSB) of Airoldi et al [1] departs from the Nowicki and Snijders [19] blockmodeling approach by instead allowing nodes to probabilistically choose the latent group. In a graph of friendships, some edges for a given node might arise from shared academic interest while other ties might arise from shared sports interest. To accomplish this, the model introduces multinomial distributions for each node, π_i, from which a latent class is drawn,

$$z_i|\pi_i \sim \text{Multinomial}(\pi_i),\ Y_{ij} \sim \text{Bernoulli}(\vec{z_i}^T \eta \vec{z_j})$$

for $1 \leq i, j \leq N$. Shafiei and Chipman [22] extended the MMSB with a mechanism for choosing the event's sender (based on their "friendliness" score) and then use the latent class assignments to determine which receivers are chosen for the email. For each email, all nodes sample a single class assignment, $z_i \sim \text{Multinomial}(\pi_i)$, akin to the MMSB.

In contrast to the block-oriented models above, the MPMM approach discussed in this paper directly models and clusters the events in the network rather than the relationships among latent groups of individuals.

One can also interpret the MPMM as a mixture model for contingency tables. Specifically, we model the marginal distributions for each domain name with a mixture of multinomials, and enforce that the classes for each dimension are linked. Similar models include the Interaction Component Model (ICMc) [23] though our model applies directly to dyadic event data and can handle directed, weighted networks with edge types. Nonparametric versions of such models are also of interest, as in Rogers et al. [20].

Sinkkonen et al [24] presented a general framework for latent-variable modeling of relational data that includes models such as the MPMM we presented in this paper—their focus was more on static co-occurence data rather than the type of event data that is of primary interest here.

9. CONCLUSION

Relational event data is increasingly common in large social network data sets. In this paper we proposed the MPMM as an interpretable and computationally tractable statistical model for analyzing such data. We illustrated how the model can be used to analyze large sets of events that have a sender, receiver, and action type, presented two algorithms for performing inference, evaluated predictive accuracy of the model on real data, and demonstrated how the model can be used to systematically handle practical issues such as missing data.

Acknowledgements

This work was supported in part by an NDSEG Graduate Fellowship (CDB) and by ONR/MURI under grant number N00014-08-1-1015 (PS). The authors would also like to thank Drew Frank for useful discussions, Charles Kemp for providing IRM code, and Jean-Pierre Eckmann for providing the email data.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, (September):1981-2014, 2008.

[2] C. Anderson, S. Wasserman, and K. Faust. Building stochastic blockmodels. *Social Networks*, 14(1-2):137–161, June 1992.

[3] M. C. Appleby. Competition in a red deer stag social group: rank, age and relatedness of opponents. *Animal Behavior*, 31:913–918, 1983.

[4] U. Brandes, J. Lerner, and T. a.B. Snijders. Networks evolving step by step: statistical analysis of dyadic event data. *2009 International Conference on Advances in Social Network Analysis and Mining*, pages 200–205, 2009.

[5] C. Butts. A relational event model for social action. *Sociological Methodology*, 38(1):155–20, 2008.

[6] J. Chang and D. Blei. Relational topic models for document networks. *Proc. of Conf. on AI and Statistics (AISTATS'09)*, 2009.

[7] J.-P. Eckmann, E. Moses, and D. Sergi. Entropy of dialogues creates coherent structures in e-mail traffic. *Proceedings of the National Academy of Sciences of the United States of America*, 101(40):14333–7, October 2004.

[8] S. E. Fienberg and S. S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192, 1981.

[9] S. Geisser and W. F. Eddy. A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365):153, March 1979.

[10] S. M. Goodreau, J. A. Kitts, and M. Morris. Birds of a feather, or friend of a friend? Using exponential random graph models to investigate adolescent social networks. *Demography*, 46:103–125, 2009.

[11] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Pro-*

ceedings of the National Academy of Sciences of the United States of America, 101 Suppl:5228–35, April 2004.

[12] P. D. Hoff. Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, October 2008.

[13] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: first steps. *Social Networks*, 5:109–137, 1983.

[14] C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[15] G. King,W Lowe. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57:617-642, 2003.

[16] G. Kossinets. Effects of missing data in social networks. *Social Networks*, 28:247–268, 2006.

[17] K. Kurihara, Y. Kameya, and T. Sato. A Frequency-based stochastic blockmodel. *Workshop on Information-Based Induction Sciences*, 2006.

[18] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.

[19] K. Nowicki and T. A. B. Snijders. Estimation and prediction of stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077– 1087, 2001.

[20] S. Rogers, A. Klami, J. Sinkkonen, M. Girolami, and S. Kaski. Infinite factorization of multiple non-parametric views. *Machine Learning*, 79(1-2):201–226, 2009.

[21] S. Ross. *Introduction to Probability Models*. Academic Press, 2006.

[22] M. Shafiei and H. Chipman. Mixed-membership stochastic block-models for transactional data. *Workshop on Analyzing Networks and Learning with Graphs (NIPS 2009)*, pages 1–8, 2009.

[23] J. Sinkkonen, J. Aukia, S. Kaski, C. Rudin, R. Schapire, and I. Daubechies. Component models for large networks. *ArXiv e-prints. arXiv:0803.1628.*, page 11-15, 2008.

[24] J. Sinkkonen, J. Aukia, S. Kaski. Infinite mixtures for multi-relational categorical data. *6th International Workshop on Mining and Learning with Graphs, Helsinki, Finland*, 2008.

[25] T. A. B. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75-100, 1997.

[26] Y. Wang and G. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8-19, 1987.