Network Event Data over Time: Prediction and Latent Variable Modeling

Padhraic Smyth

University of California, Irvine

Machine Learning with Graphs Workshop, July 25th 2010

Acknowledgements

- **PhD students:**
  - Arthur Asuncion, Chris DuBois, Jimmy Foulds

- **Funding**
  - National Science Foundation
  - Office of Naval Research (MURI grant)
  - NDSEG Graduate Fellowship
  - Yahoo!, Google, IBM, Microsoft, Experian

Resources

A survey of statistical network models

A. Goldenberg, A. Zheng, S. Fienberg, E. Airoldi, *Foundations and Trends in Machine Learning*, 2009

Multiplicative latent factor models for description and prediction of social networks

P. D. Hoff, *Computational and Mathematical Organization Theory , 2009.*

Random effects models for network data

P. D. Hoff, in *Dynamic Social Network Modeling and Analysis, 2003*

A relational event model for social action

C. E. Butts, *Sociological Methodology*, 2008

Slides from 2010 Whistler Summer School on Social Networks

http://people.cs.ubc.ca/~murphyk/pims2010Whistler/

Static Network Data

- **General Notation:**
  - N actors (node set)
    - Will assume that set of actors is known and fixed
  - Edges between actors (Y)
    - Adjacency matrix Y
    - $y_{i,j}$ indicates an edge between actor i and actor j
    - Simplest case: binary undirected/directed edges
  - Covariates/Attributes (X)
    - e.g., for each actor (e.g., age, text documents,..)
    - e.g., for each edge (e..g., numeric weights, vector of attributes, text, etc)

Dynamic Network Data

Case 1: discrete-time

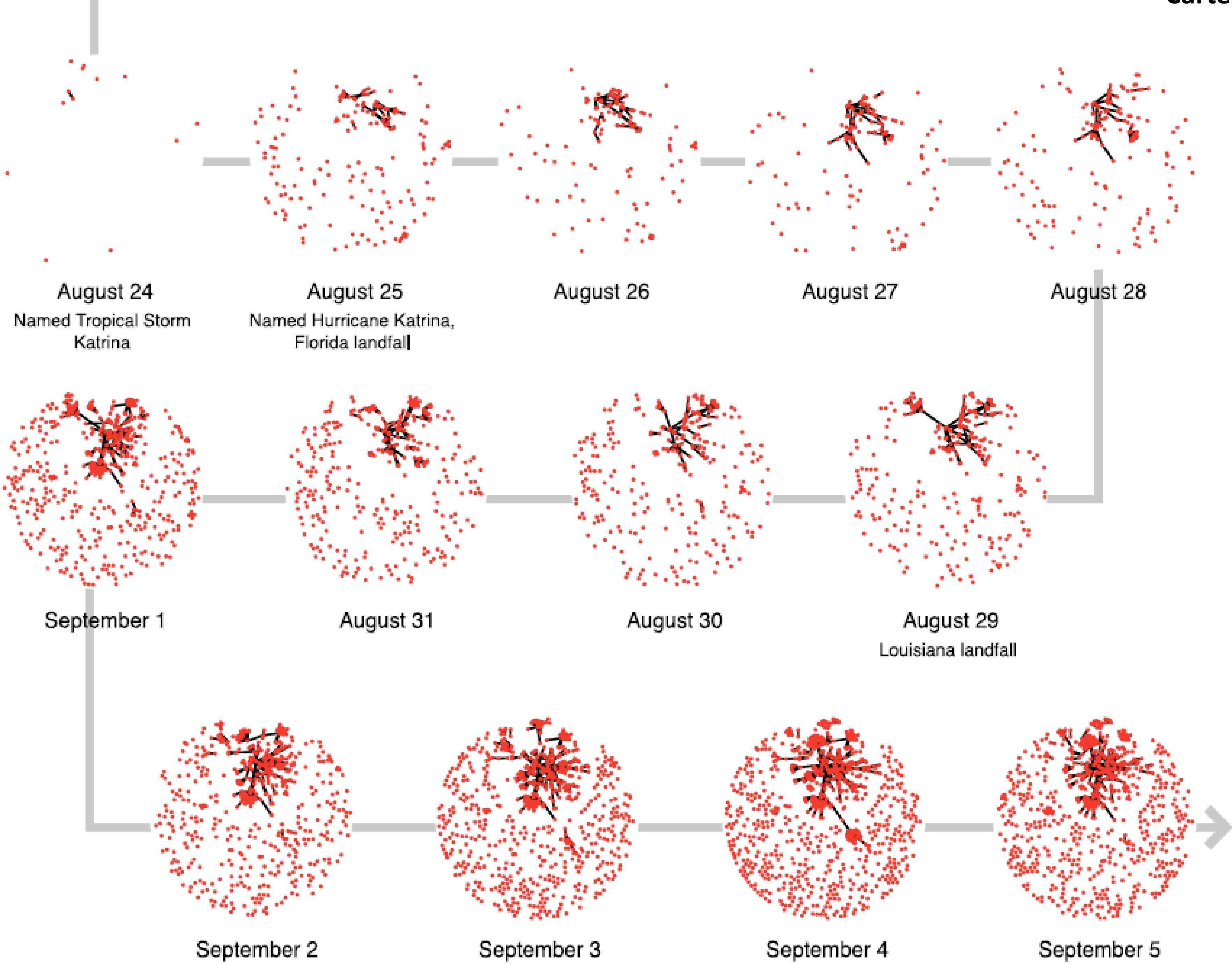
- Y_t represents the state of the network at discrete time t
- Data D = $\{Y_1 \ldots\ldots Y_t \ldots\ldots\ldots Y_T\}$

Example

- actors = students in a school
- Y_t = friendships between students in month t, t = 1, … 12

Interest is often in network dynamics and evolution

e.g., Markov models for $P(Y_{t+1} \mid Y_t)$


August 24
Named Tropical Storm Katrina
August 25
Named Hurricane Katrina, Florida landfall
August 26
August 27
August 28
September 1
August 31
August 30
August 29
Louisiana landfall
September 2
September 3
September 4
September 5

Dynamic Network Data

Case 2: continuous-time network events

- y_t is an edge between some pair i and j at time t
- Birth-death edges: each y_t has a start and end time
- Flipping edges: edges can switch on or off
- Instantaneous edges: each y_t is (effectively) instantaneous
- Data D = { y_1 …… y_t ……… y_T } - in a sense there is no graph

Example

- actors = students in a school
- y_t = email between 2 students at time t
  (would need to allow for multiple recipients…)

Interest is often in rates and patterns of communication
e.g., Poisson rates for $y_{i,j}$ given network history up to time t

Enron Email Data

(Figure from Goldenberg et al, 2010)

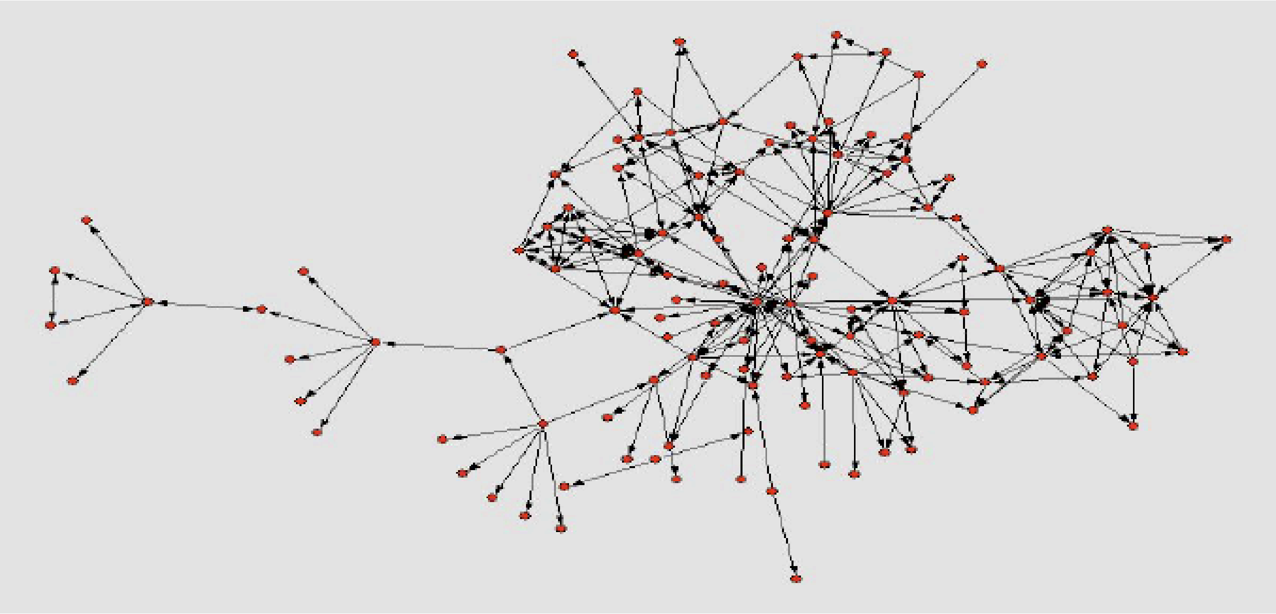
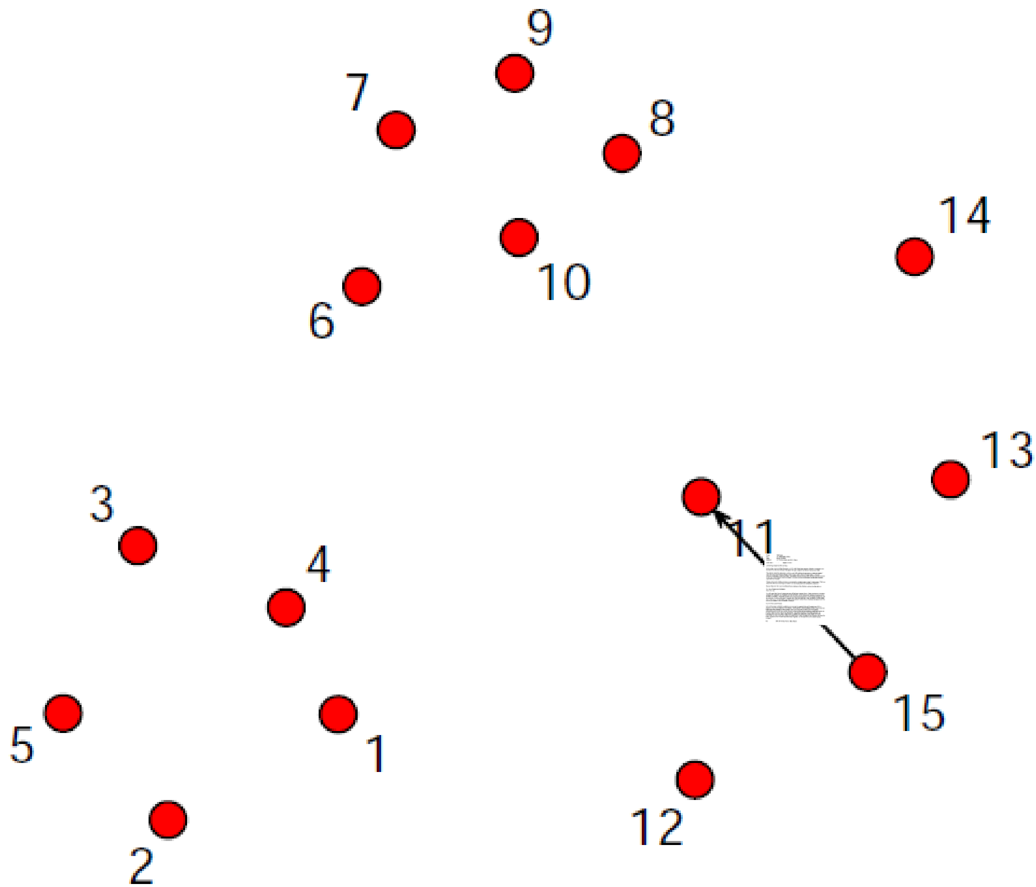


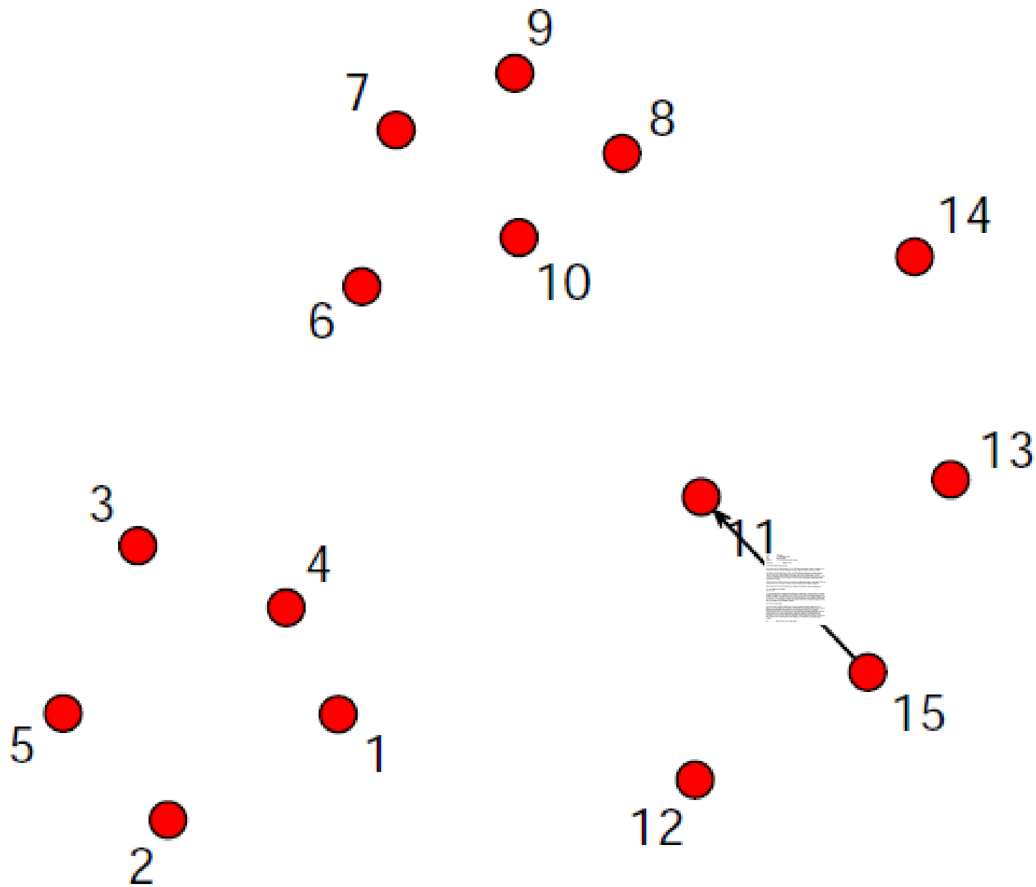
Figure 2.3: E-mail exchange data among 151 Enron executives, using a threshold of a minimum of 30 messages for each link. Source: [153].

Time 1

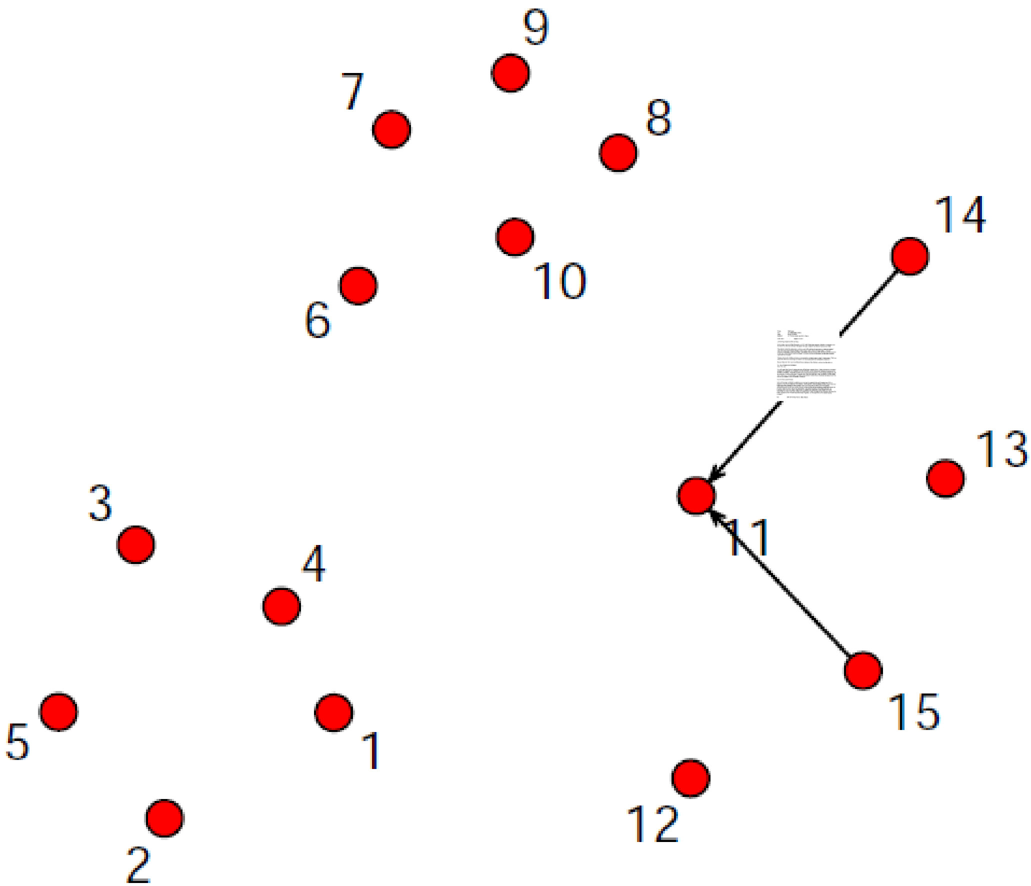


UCIRVINE | UNIVERSITY of CALIFORNIA

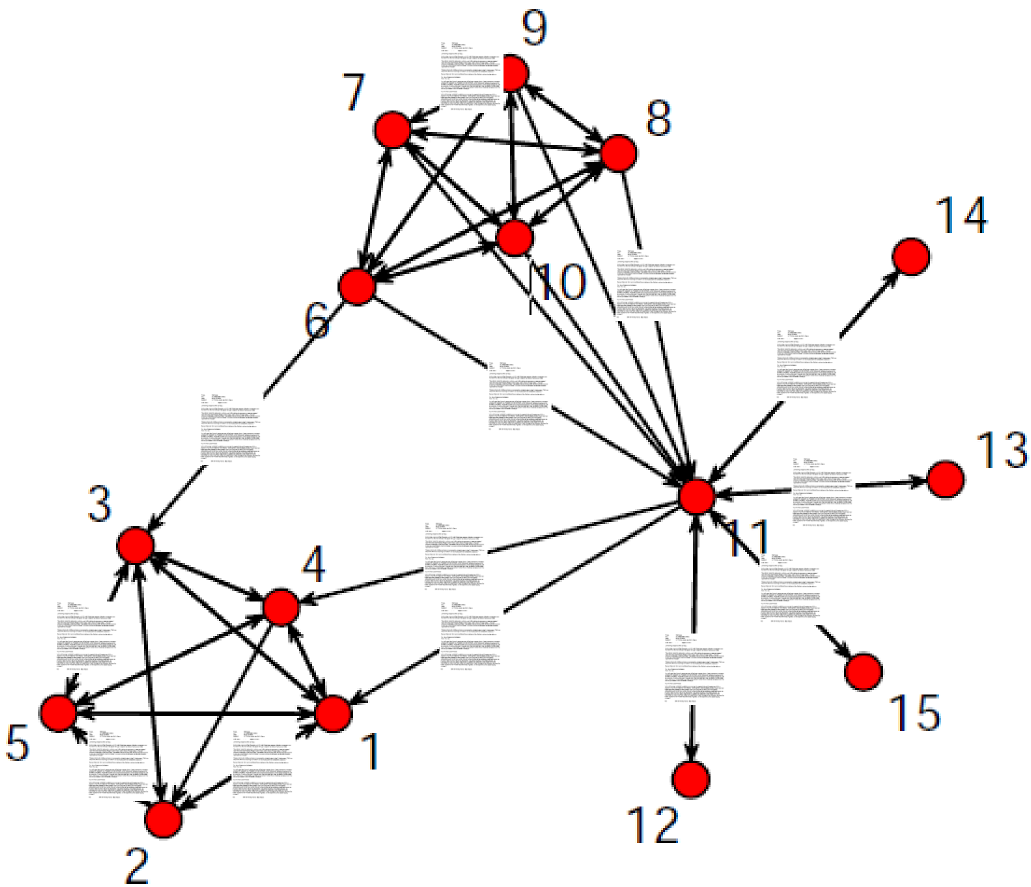
Time 1



Time 2



Time 50



Relational Event Model

Butts, 2009

λ (i,j) = Poisson rate of edge generation between actor i and actor j

λ^t (i,j) = function of network features up to time t,

Results in a piecewise constant inhomogeneous Poisson process

- rates(t) are a function of network history at time t
- between events the rates are constant

Typical features include:

- individual actor effects
- persistence between pairs
- preferential attachment
- conversational behavior

Relational Event Model

- **Example**

$$\log \lambda^t(i,j) = \log \lambda_0 + \log \lambda_i + \log \lambda_j + \underbrace{\beta^t}_{\text{p-dim vector of weights}} \; \underbrace{x^t(i,j)}_{\text{p-dim vector of features}}$$

- **Estimation**
  - Can be fit with standard regression methods (survival analysis)
  - Likelihood involves $O(N^2)$ terms for each of T events
    - Does not scale well
  - Nonetheless an interesting model....
    - See Butts (2008) for an application to emergency response communications

Ordinal Version of Relational Events

- If we don't have time-stamps, but do have the order of the events…
- Can use the fact that "choice probability" can be written as

$$P(i, j) = \lambda^t (i,j) \;/\; \Sigma \; \lambda^t (i,j)$$

- Can still learn the model from sequence of events, with relative rates
  - Overall network rate λ_0 is unspecified

Additional Modeling Aspects….

- **As with static networks….**
  - Actor attributes, e.g., actor age
  - Edge (event) attributes, e.g., text of an email

- **Can also have time-dependent covariates/attributes**
  - E.g., actor attributes changing over time
  - Network level "external" covariates
    - Calendar effects: time of day, day of week, time of year
    - External events – exogenous time-series

Outline of the Talk

- **Begin with statistical models for static data**
  - In particular, latent variable models
  - Review some useful approaches in this area

- **Look at how to extend these models to temporal data**
  - Particularly relational event data
  - Discuss recent work

    [Caveat: only focus on certain approaches, not exhaustive]

- **Evaluation and prediction**
  - Some general comments

- **Mostly review….with some new work towards the end**

Why Statistical Modeling?

- **Learning**
  - can estimate network properties from data in a principled way
- **Prediction/Querying**
  - reduces to computation of relevant conditional probabilities and expectations
- **Noise/Missing Data**
  - Systematic way to handle real-world noise
- **Covariates**
  - Relatively straightforward to integrate “non-network” information into the network model

Slide from Dave Hunter

Exponential-family Random Graph Model (ERGM)

$$P_\eta(Y = y) \propto \exp\{\eta^t g(y)\}, \qquad y \in \mathcal{Y}$$

or

$$P_\eta(Y = y) = \frac{\exp\{\eta^t g(y)\}}{\kappa(\eta)}, \qquad y \in \mathcal{Y}$$

where

- η is a vector of parameters
- $g(y)$ is a known vector of graph statistics on y
- $\kappa(\eta)$ is the normalizing constant:

$$\kappa(\eta) = \sum_{z \in \mathcal{Y}} \exp\{\eta^t g(z)\}$$

Estimation is Hard

$P(G \mid \theta) = f(G ; \theta) /$ *normalization constant*

The normalization constant = sum over all possible graphs

Say binary directed graphs: how many graphs? $2^{n(n-1)}$

e.g., with $n = 50$, we will have $2^{2450} \sim 10^{245}$ graphs to sum over

MCMC techniques are now the method of choice – but many problems with degeneracy of likelihoods – difficult models to fit

(e.g., see Robins et al, Social Networks, 2007)

An Alternative: Latent Variable Models

- **ERGMs**
  - allow us to model edge dependencies in very flexible ways – but the computational penalty is too high

- **Latent Variable Models**
  - Typically, latent variables are chosen so that edges are conditionally independent given the latent variables
    - Can lead to much simpler models than full ERGMs

  - If we can find useful and tractable latent variable representations, this may provide a good alternative to ERGMs

Example: The Latent Space Model

Hoff, Raftery, Handcock, 2002

- **Idea:**
  - Embed nodes in a latent K-dimensional Euclidean space
  - Probability of edge (i,j) = f (distance (i, j))
  - Edges are conditionally independent given K-dim locations

Example: The Latent Space Model

Hoff, Raftery, Handcock, 2002

- **Idea:**
  - Embed nodes in a latent K-dimensional Euclidean space
  - Probability of edge (i,j) = f (distance (i, j))
  - Edges are conditionally independent given K-dim locations

- **Probability model**
  - z_i = K-dim latent position vector for node i

  Log-odds (y_{ij} = 1) = log P(y_{ij} = 1)/(1 –P(y_{ij} = 1))

$$= -\,|\, z_i - z_j \,| + \mu + \beta x_{ij}$$

distance of nodes i and j

network density parameter

covariate effects (optional)

Example: The Latent Space Model

Hoff, Raftery, Handcock, 2002

- **Likelihood:**

$$P(Y \mid Z, b, m) = \prod P(y_{ij} \mid z_i, z_j, \mu, \beta)$$

logistic function

- **Estimation:**
  - Can maximize likelihood directly (as a function of Z,….) using gradient methods
  - Can also be Bayesian, use priors, and sample from posterior density using MCMC
  - Can also introduce block/cluster structure on nodes
    (see Handcock, Raftery, and Tantrum, 2007)

- **Computational issues**
  - Note that the product above is over all pairs, $O(N^2)$: poor scalability
  - Recent work (Raftery et al, 2010) shows how to ignore many non-edges

Figure from Hoff, Raftery, Handcock, 2002

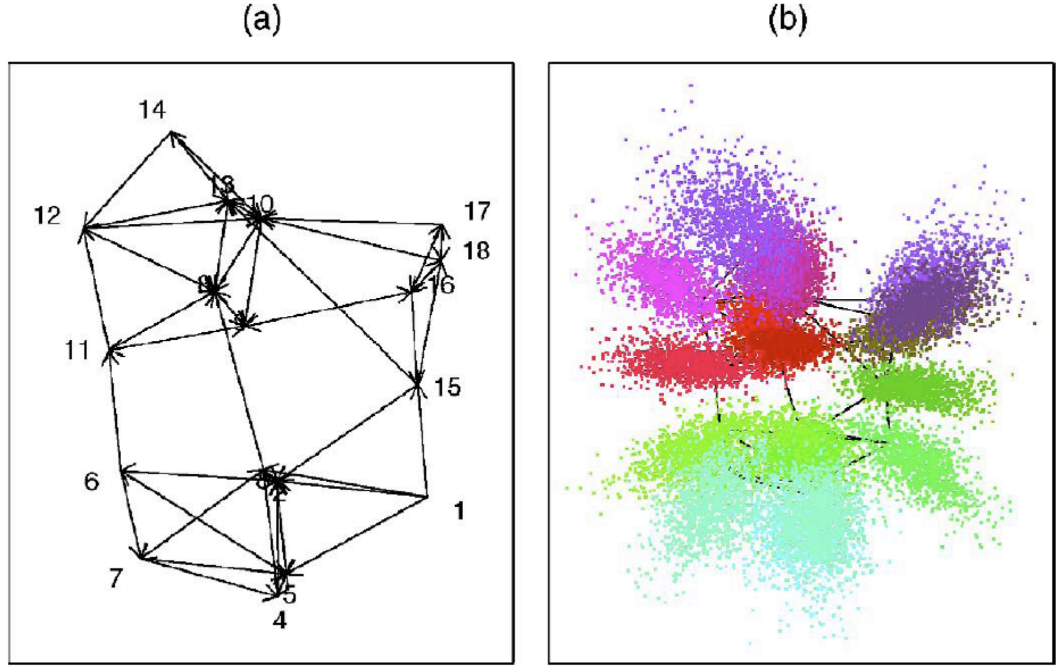


Figure 1. Maximum Likelihood Estimates (a) and Bayesian Marginal Posterior Distributions (b) for Monk Positions. The direction of a relation is indicated by an arrow.

Figure from Hoff, Raftery, Handcock, 2002

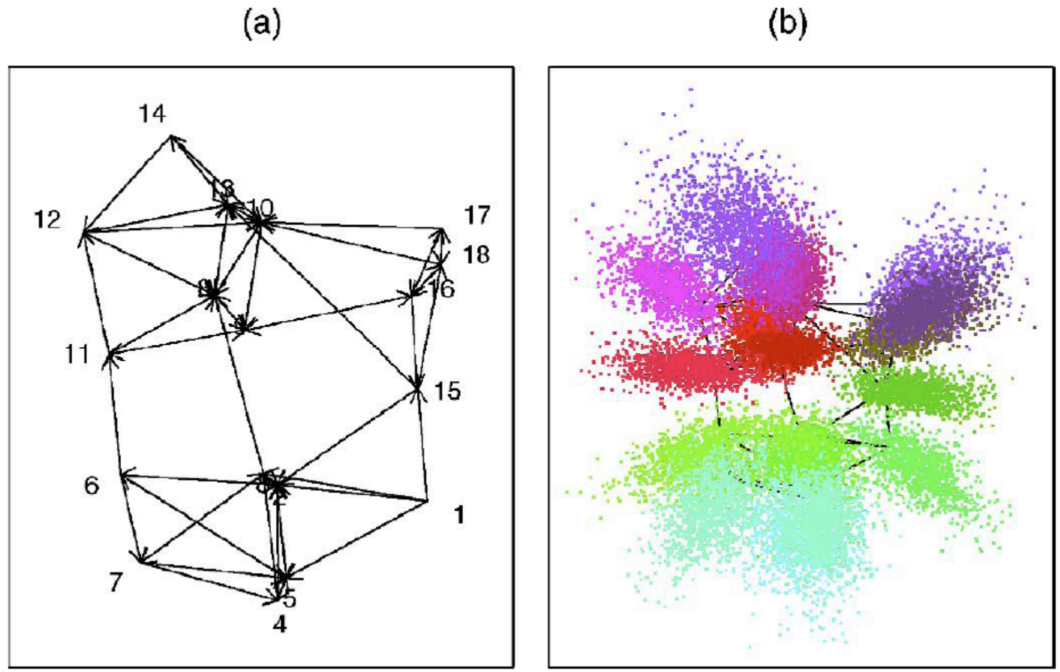


Figure 1. Maximum Likelihood Estimates (a) and Bayesian Marginal Posterior Distributions (b) for Monk Positions. The direction of a relation is indicated by an arrow.

- **Representational issue**
  - Is Euclidean space embedding a good way to represent network information? Similar to issues with multidimensional scaling

Example: Relational Topic Model

Chang and Blei, 2009

- **Nodes = documents, edges = links between documents**

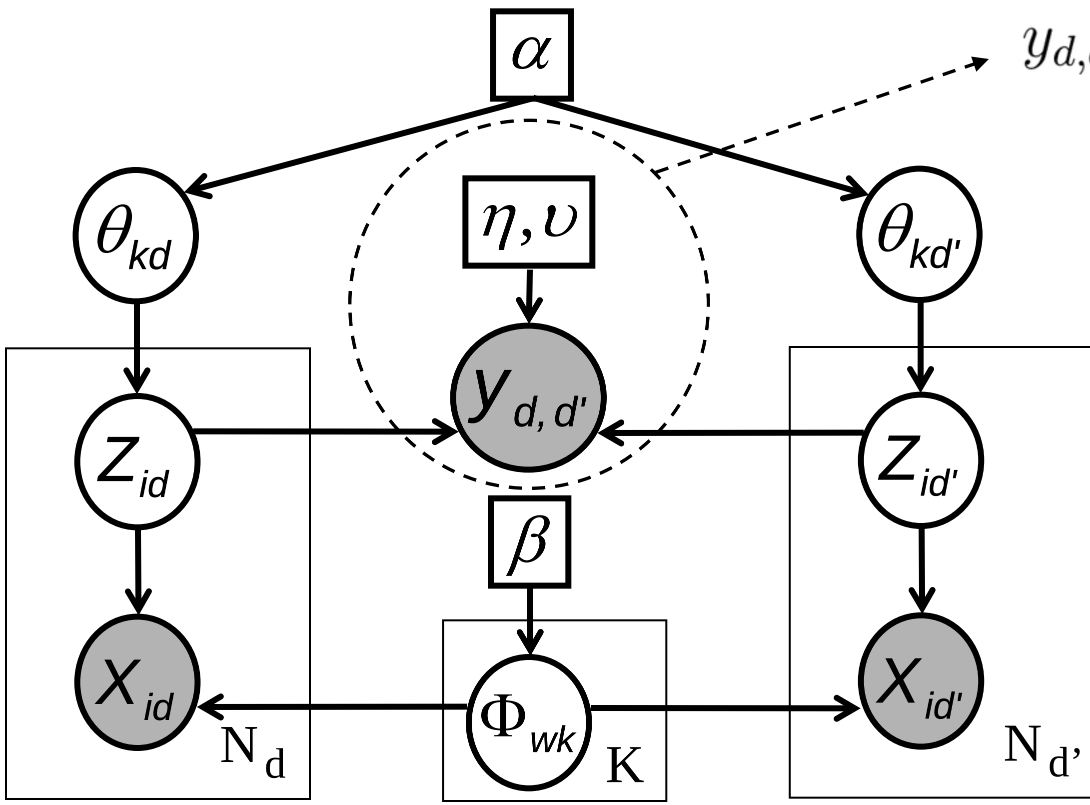
- **"Standard" LDA/topic model, but ..**
  - Topics i,j influence p(edge i, j)
  - Edge(i, j) influences topics i and j

- **Model is similar to latent-space model**
  - Latent space: actors represented by k-dim location
  - Relational topics: docs represented by k-dim topic distribution
  - Both use logistic-like links for edge probabilities

Relational Topic Model (RTM)

[Chang, Blei, 2009]

- **Same setup as LDA, except we have observed network information across documents (adjacency matrix)**



$$y_{d,d'} \sim \psi(y_{d,d'} | \mathbf{z}_d, \mathbf{z}_{d'}, \eta, \nu)$$

"Link probability function"

Documents with similar topics are more likely to be linked

Topics influence links, and links infuence topics

Link probability functions

- **Exponential:** $\psi(y_{d,d'} = 1) = \exp(\eta^T(\bar{\mathbf{z}}_d \circ \bar{\mathbf{z}}_{d'}) + \nu)$

  K-dim vector of topic proportions

- **Logistic:** $\psi(y_{d,d'} = 1) = \sigma(\eta^T(\bar{\mathbf{z}}_d \circ \bar{\mathbf{z}}_{d'}) + \nu)$

- **Normal CDF:** $\psi(y_{d,d'} = 1) = \Phi(\eta^T(\bar{\mathbf{z}}_d \circ \bar{\mathbf{z}}_{d'}) + \nu)$

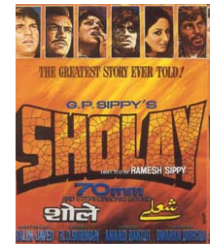
- **Normal:** $\psi(y_{d,d'} = 1) = \exp(-\eta^T(\bar{\mathbf{z}}_d - \bar{\mathbf{z}}_{d'}) \circ (\bar{\mathbf{z}}_d - \bar{\mathbf{z}}_{d'}) - \nu)$

  Element-wise product

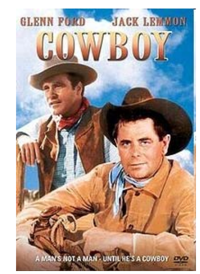
  - where $\bar{\mathbf{z}}_d = \frac{1}{N_d}\sum_i z_{id}$

Link Prediction with Wikipedia Movie Pages

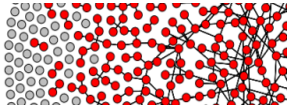
- **'Sholay'**
  - Indian film, 45% of words belong to topic 24 (Hindi topic)
  - Top 5 most probable movie links in training set:
    - 'Laawaris'
    - 'Hote Hote Pyaar Ho Gaya'
    - 'Trishul'
    - 'Mr. Natwarlal'
    - 'Rangeela'



- **'Cowboy'**
  - Western film, 25% of words belong to topic 7 (western topic)
  - Top 5 most probable movie links in training set:
    - 'Tall in the Saddle'
    - 'The Indian Fighter'
    - 'Dakota'
    - 'The Train Robbers'
    - 'A Lady Takes a Chance'



- **'Rocky II'**
  - Boxing film, 40% of words belong to topic 47 (sports topic)
  - Top 5 most probable movie links in training set:
    - 'Bull Durham'
    - '2003 World Series'
    - 'Bowfinger'
    - 'Rocky V'
    - 'Rocky IV'



Example: Stochastic Block Model

e.g., Nowicki and Snijders, 2001

- **Idea:**
  - Partition the set of nodes into K “blocks” that are “structurally equivalent”
  - Model interactions at the K x K block level instead of N x N actor level

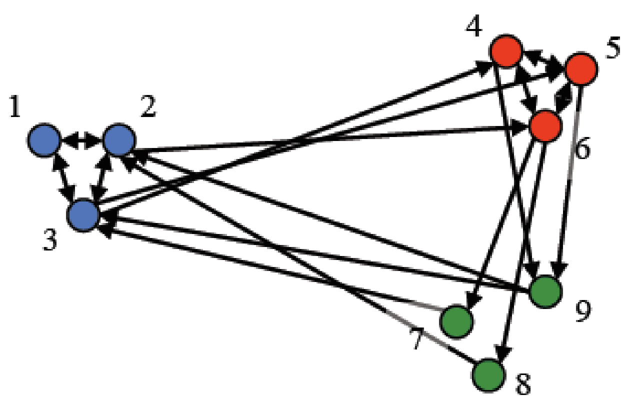
$$P(y_{ij}) = P(y_{k_i, k_j}), \quad k_i, k_j \in \{1, \ldots\ldots K\}$$

Example: Stochastic Block Model

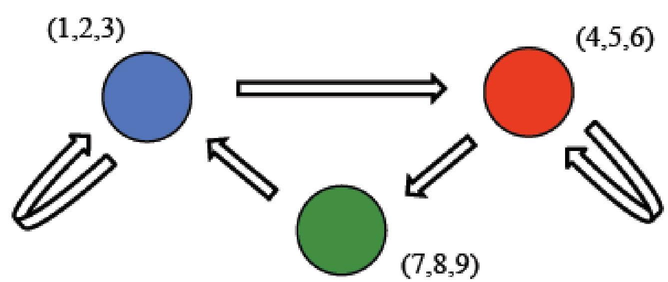
e.g., Nowicki and Snijders, 2001

- **Idea:**
  - Partition the set of nodes into K “blocks” that are “structurally equivalent”
  - Model interactions at the K x K block level instead of N x N actor level

$$P(y_{ij}) = P(y_{ki,\, kj}), \qquad k_i, k_j \;\varepsilon\; \{1,......K\}$$



(Figure from Goldenberg et al, 2010)



Example: Stochastic Block Model

e.g., Nowicki and Snijders, 2001

- **Estimation**
  - 2 sets of parameters

    1. B = block-level interaction matrix, e.g., K x K matrix of Bernoullis

    2. Z = N indicator variables, mapping each node to one of K blocks

  (Can use your favorite estimation technique: EM, gradient, MCMC, etc)

  - See also Infinite Relational Model (IRM), Kemp et al (2006)
    - Allows one to learn the number of blocks

Mixed Membership Stochastic Block Model (MMB)

Airoldi, et al, 2008

- **Generalizes the stochastic block model to allow mixed membership**

- **Specifically,**
  - Replace N indicator variables, with N multinomials z_i , i = 1, …N
  - Each multinomial is a distribution over the K blocks
    - Allows an actor to have multiple memberships, with prob z_{i1}, … z_{iK}

Mixed Membership Stochastic Block Model (MMB)

Airoldi, et al, 2008

- **Generalizes the stochastic block model to allow mixed membership**

- **Specifically,**
  - Replace N indicator variables, with N multinomials z_i , $i = 1, \ldots N$
  - Each multinomial is a distribution over the K blocks
    - Allows an actor to have multiple memberships, with prob $z_{i1}, \ldots z_{iK}$

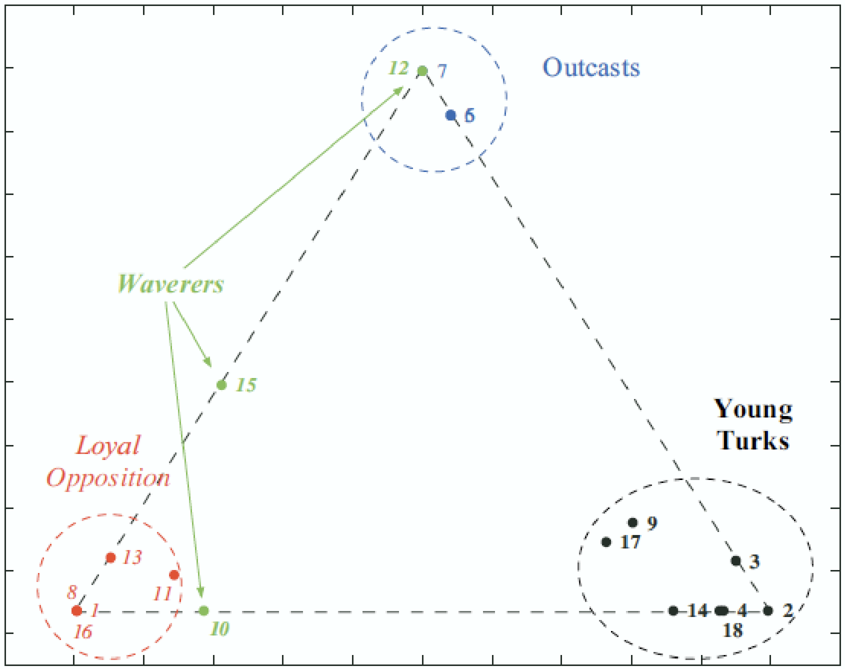
- **Generative model**
  - For each actor: multinomial z_i ~ Dirichlet
  - For each possible edge:

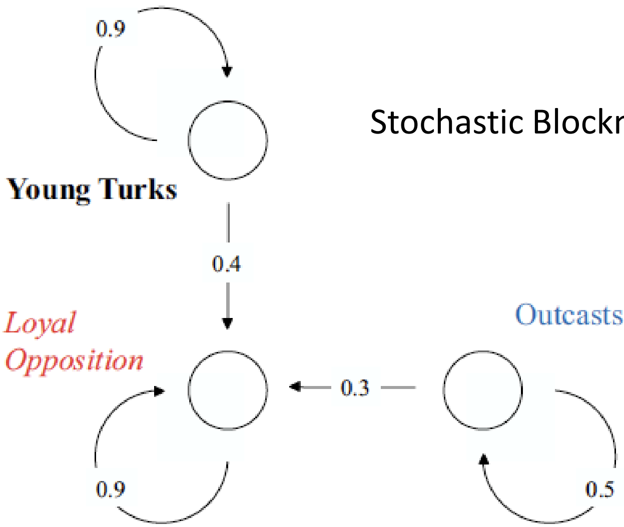
$$k_i \sim \text{multinomial } z_i \quad , \quad k_j \sim \text{multinomial } z_j$$

$$y_{ij}, \sim B(k_i , k_j)$$

- **Estimation**
  - Likelihood involves $O(N^2)$ terms: can use variational or MCMC methods

Mixed Membership Stochastic Blockmodel





Stochastic Blockmodel

Figures from Airoldi, et al, 2008

Binary Feature Relational Model

Miller, Griffiths, Jordan, 2009

- **Based on idea of Indian Buffet Process (Griffiths and Ghahramani, 2006)**
  - Represent each object by a set of latent binary features
  - Learn binary features that explain well the observed data
  - Non-parametric: infinite number of features

    ….but in practice, given data, only a finite number are inferred

- **Motivation:**
  - Classes defined over combinatorial number of binary features
  - Different from MMB, e.g., “male high school musicians/athletes”
  - Different from latent space

- **Can apply this idea to network data**
  - Latent variable model where p(edge i, j) is a function of i and j’s latent binary features

Relational Binary Feature Model for Networks

Miller, Griffiths, Jordan, 2009

Hidden Features

Actors

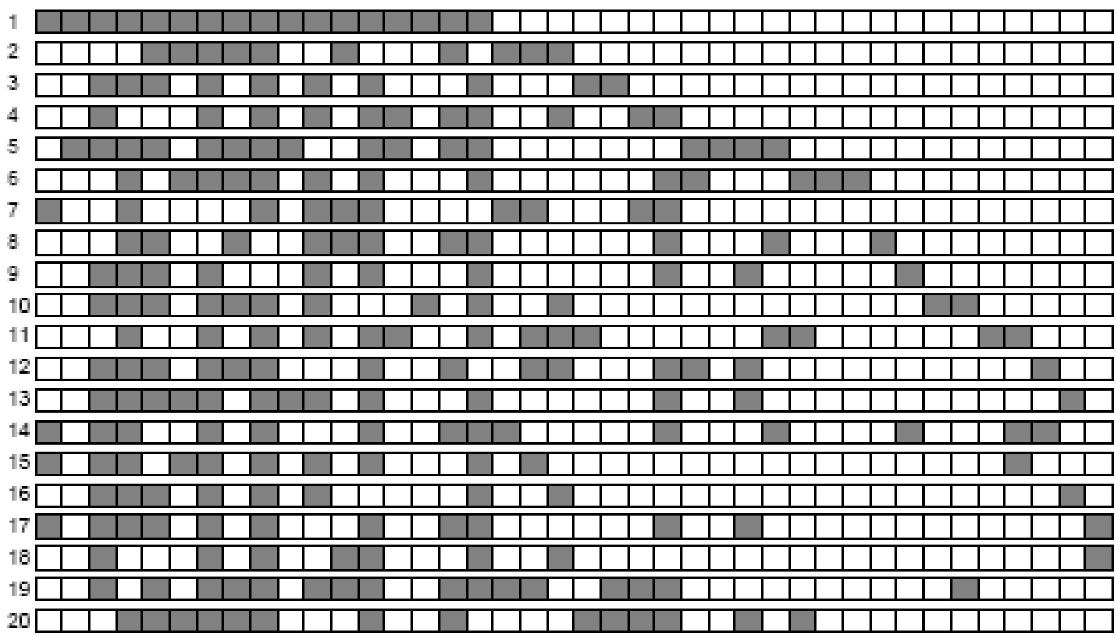
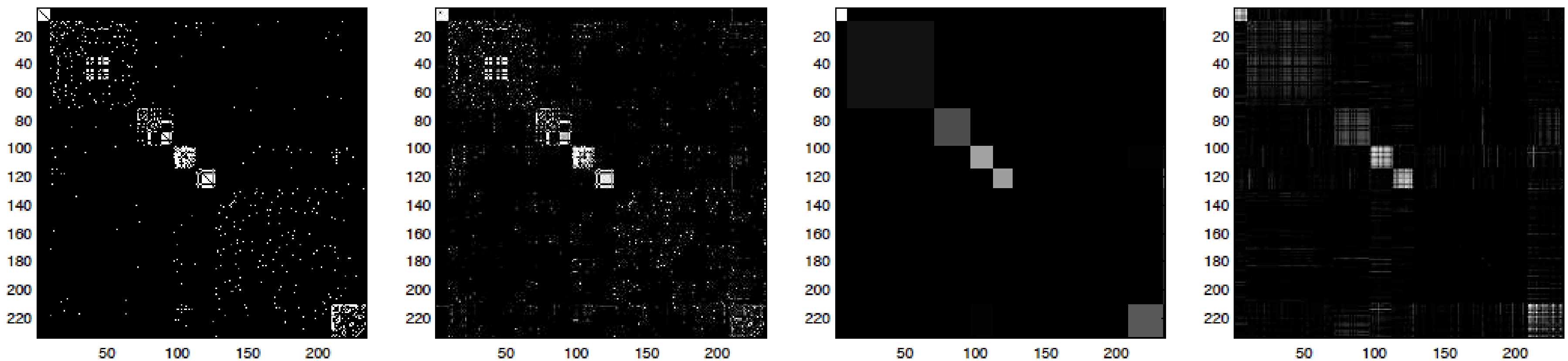


Figure from Griffiths and Ghahramani, 2006

Presence of edge between actor i and actor j is (e.g.)
a logistic function of a weighted sum of features they have in common

Estimation: based on MCMC

Predictions on NIPS Coauthorship Data



(a) True relations (b) Feature predictions (c) IRM predictions (d) MMSB predictions

Figure 2: Predictions for all algorithms on the NIPS coauthorship dataset. In (a), a white entry means two people wrote a paper together. In (b-d), the lighter an entry, the more likely that algorithm predicted the corresponding people would interact.

From Miller, Griffiths, Jordan, 2009

A Unified View…

(see also Hoff, 2009; Airoldi 2010)

$$P(y_{ij} = 1) = f\left(g(z_i, z_j) + \beta x_{ij} + \mu \right)$$

where

- f = logistic function (for example)
- z_i , z_j = k x 1 latent vectors for the ith and jth nodes
- g = function that combines latent vectors, with parameters θ
- x_{ij} = covariate vector for the pair of nodes
- μ = network density parameter

Examples $\text{Log-odds}(y_{ij} = 1) = g(z_i, z_j) + \beta x_{ij} + \mu$

Latent space model:

z_i, z_j = k x 1 vectors of latent positions in Euclidean space

$g(z_i, z_j) = -|z_i - z_j|$

Examples

$$\text{Log-odds}(y_{ij} = 1) = g(z_i, z_j) + \beta x_{ij} + \mu$$

Latent space model:

z_i , z_j = k x 1 vectors of latent positions in Euclidean space

$g(z_i, z_j) = - | z_i - z_j |$

Latent factor model: (see Hoff, 2008)

z_i = k x 1 real-valued vector

$g(z_i, z_j) = z'_i W z_j$, where W is a k x k diagonal matrix

Examples Log-odds ($y_{ij} = 1$) = $g(z_i, z_j) + \beta x_{ij} + \mu$

Latent space model:

z_i , z_j = k x 1 vectors of latent positions in Euclidean space

$g(z_i, z_j) = - | z_i - z_j |$

Latent factor model: (see Hoff, 2008)

z_i = k x 1 real-valued vector

$g(z_i, z_j) = z'_i W z_j$, where W is a k x k diagonal matrix

Relational topic model:

z_i = k-dimensional topic distribution (multinomial) for document i

$g(z_i, z_j)$ = weighted element-wise product of the 2 topics

Examples

$\text{Log-odds}(y_{ij} = 1) = g(z_i, z_j) + \beta x_{ij} + \mu$

Latent class or stochastic blockmodel:

z_i = fixed k-dimensional binary indicator vector, e.g., (0, 0, 1, 0 , 0)

$g(z_i, z_j) = W_{zi, zj}$, where W is a k x k matrix

The indicators select which element (block) to use

Examples

Log-odds (y_{ij} = 1) = $g(z_i , z_j) + \beta x_{ij} + \mu$

Latent class or stochastic blockmodel:

z_i = fixed k-dimensional binary indicator vector, e.g., (0, 0, 1, 0 , 0)

$g(z_i , z_j) = W_{zi , zj}$, where W is a k x k matrix

The indicators select which element (block) to use

Mixed membership stochastic blockmodel (MMB)

Like latent class, but z_i = sampled from "actor multinomial" i

Examples

Log-odds (y_{ij} = 1) = g(z_i , z_j) + β x_{ij} + μ

Latent class or stochastic blockmodel:

z_i = fixed k-dimensional binary indicator vector, e.g., (0, 0, 1, 0 , 0)

g(z_i , z_j) = $W_{zi\,,\,zj}$, where W is a k x k matrix

The indicators select which element (block) to use

Mixed membership stochastic blockmodel (MMB)

Like latent class, but z_i = sampled from “actor multinomial” i

Relational binary feature model (finite version):

z_i = k-dimensional binary vector, e.g., (1, 0, 1, 0 , 1)

g(z_i , z_j) = z'_i W z_j , where W is a k x k matrix

The combination of “on” features determine the pairwise effec

Adding Time....

General static form for latent variable models:

$$\text{Log-odds}(y_{ij} = 1) = g(z_i, z_j) + \beta x_{ij} + \mu$$

One approach is to make the z's time-dependent

i.e., allow latent features of each actor change over time

An example: Gaussian linear motion models in z-space

- Sarkar and Moore (2005) for actors' latent-space positions
- Fu, Song, and Xing (2009) for actors' mixed membership vectors

Event Data and Latent Variables

Data = series of time-stamped binary directed events among actors

2 processes we want to model:

- Rates (e.g., Poisson)
- Choices (who connects to who)

A simple approach:

Each pair i, j (at time t) has an event rate that is Poisson λ_{ij}

Global network rate = $\Sigma \ \lambda_{ij} = \lambda$

$P(y_{ij}) = \lambda_{ij} / \lambda$ or, $\lambda_{ij} = P(y_{ij}) \times \lambda$

Here $P(y_{ij})$ is a multinomial with $O(N^2)$ entries : given that an event will happen, which pair will it be?

(different from binary y_{ij} variables we saw before)

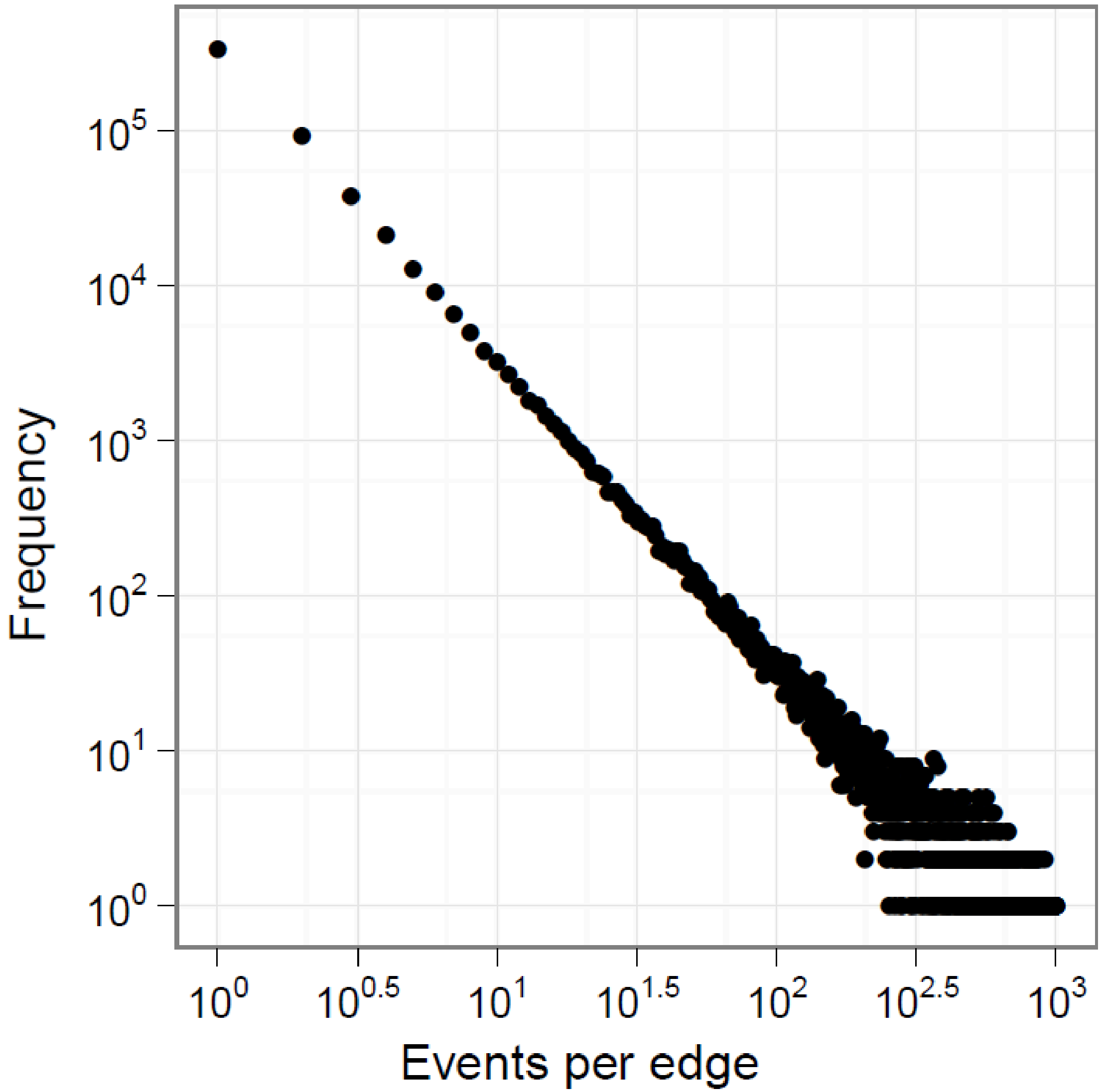
Direct Estimation

- **We could predict the likelihood of i and j communicating based directly on i and j's history**
  - Multinomial with $O(N^2)$ entries
  - Can use smoothing to combat sparsity

- **Problems**
  - Data can be extremely sparse for large N – smoothing is non-informative, and does not "borrow strength" from the graph

- **Nonetheless this is a useful baseline when evaluating predictions**
  - Historically, few papers evaluate models predictively
  - Even fewer compare their models to simple baselines

Illustration of Sparsity: Frequency of Events per pair of Actors



International Political Events data
King, 2003

Mixtures for Relational Events

Talk by Chris DuBois, Tuesday

- **Mixture model over events**
  - First choose event class k, $k = 1, \ldots. K$
    - $k \sim \pi$
    - y_{ij} : $i \sim \phi$ (sender nodes | k), $j \sim \phi$ (receiver nodes | k)
  - Parameters
    - π : K x 1 multinomial = relative likelihood of different event classes
    - ϕ (sender nodes | k), ϕ (receiver nodes | k)
      - 2K multinomials, each of size N

- **Simple model**
  - Similar to model proposed by Sinkkonen et al, MLG 2008
  - Quite similar in spirit to LDA/topic model for documents

Marginal Product Mixture Model (MPMM)

- **Likelihood**

$$P(D|\Phi) = \prod_{i=1}^{T} \sum_{c=1}^{C} P(s_i|\theta, c) P(r_i|\phi, c) P(a_i|\psi, c) P(c|\pi)$$

$$= \prod_{i=1}^{T} \sum_{c=1}^{C} \theta_{c,s_i} \phi_{c,r_i} \psi_{c,a_i} \pi_c$$

- **Estimation**
  - Can use EM or Collapsed Gibbs Sampling
    - Both are fast – only need to loop over observed events (can ignore pairs where no events occurred)

- **Extensions**
  - Modulate "choice process" with time-varying network rate
  - Different types of events ("actions")
  - Markov (hidden) dependence on selection of event class

Comparing MPMM and MMB

- **MMB model**
  - For every pair of actors
    - Sample latent class for i and j
    - Given latent classes, sample a binary edge, or a count (e.g., Poisson)

- **MPMM model**
  - For every event
    - Select latent class of event
    - Given latent class, sample i and j

- **Differences**
  - MMB models whole graphs, but not individual events
    - So dynamics are from graph to graph (e.g., Fu, Song, Xing 2009)
  - MPMM models individual events, not whole graphs
    - Allows dynamics at the event level (e.g., Markov dependence of events)
  - And inference in MPMM is much more tractable….

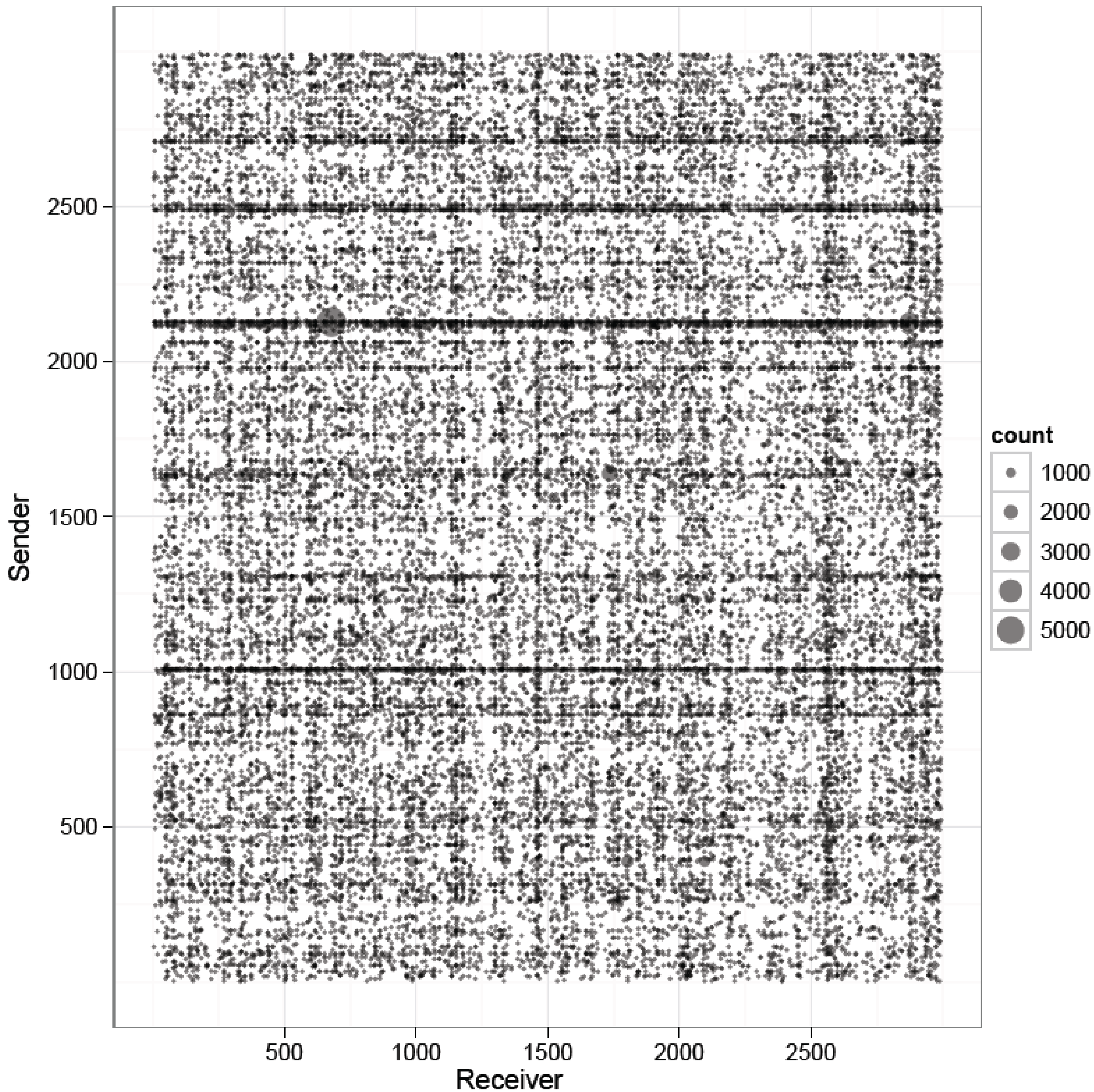
Estimation

- **EM**
  - Straightforward and fast
- **MCMC, Collapsed Gibbs sampling**
  - Also straightforward and fast
- **Both EM and Gibbs scale linearly in the number of observed events (edges)**
  - Easy to apply to large data sets

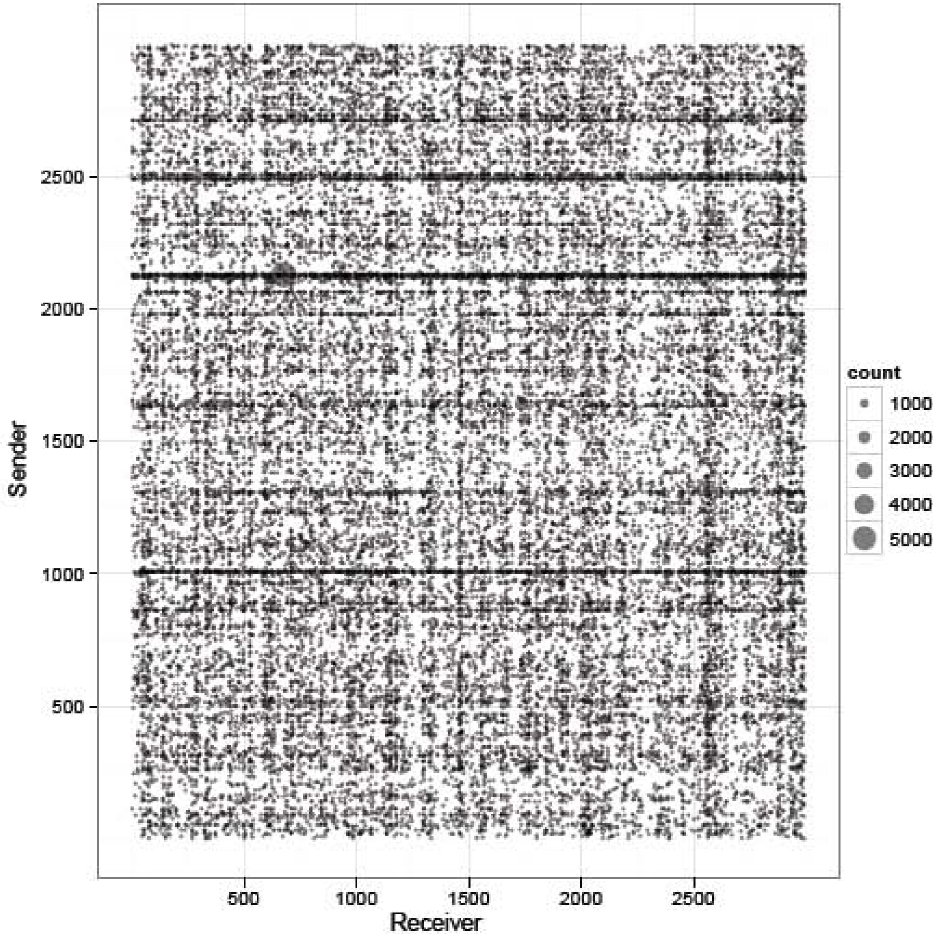
Eckmann Email Data Set

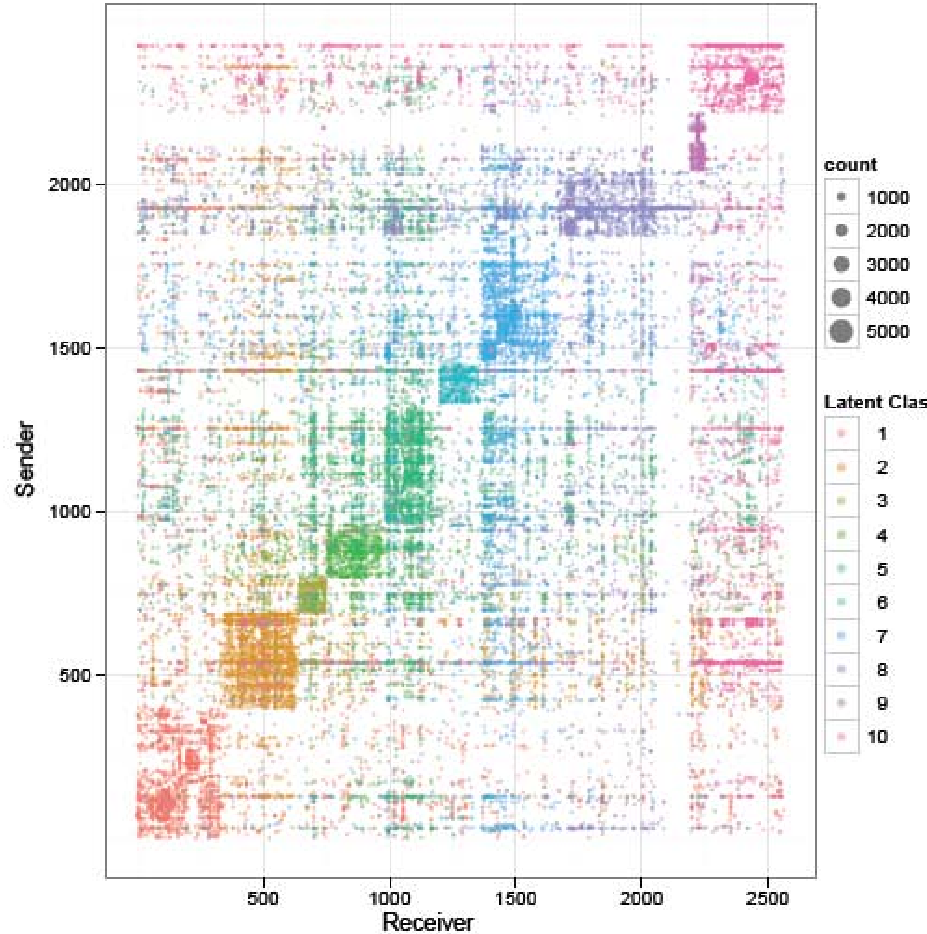
200,000 emails

2997 individuals, 82 days



Email Data (Eckmann)



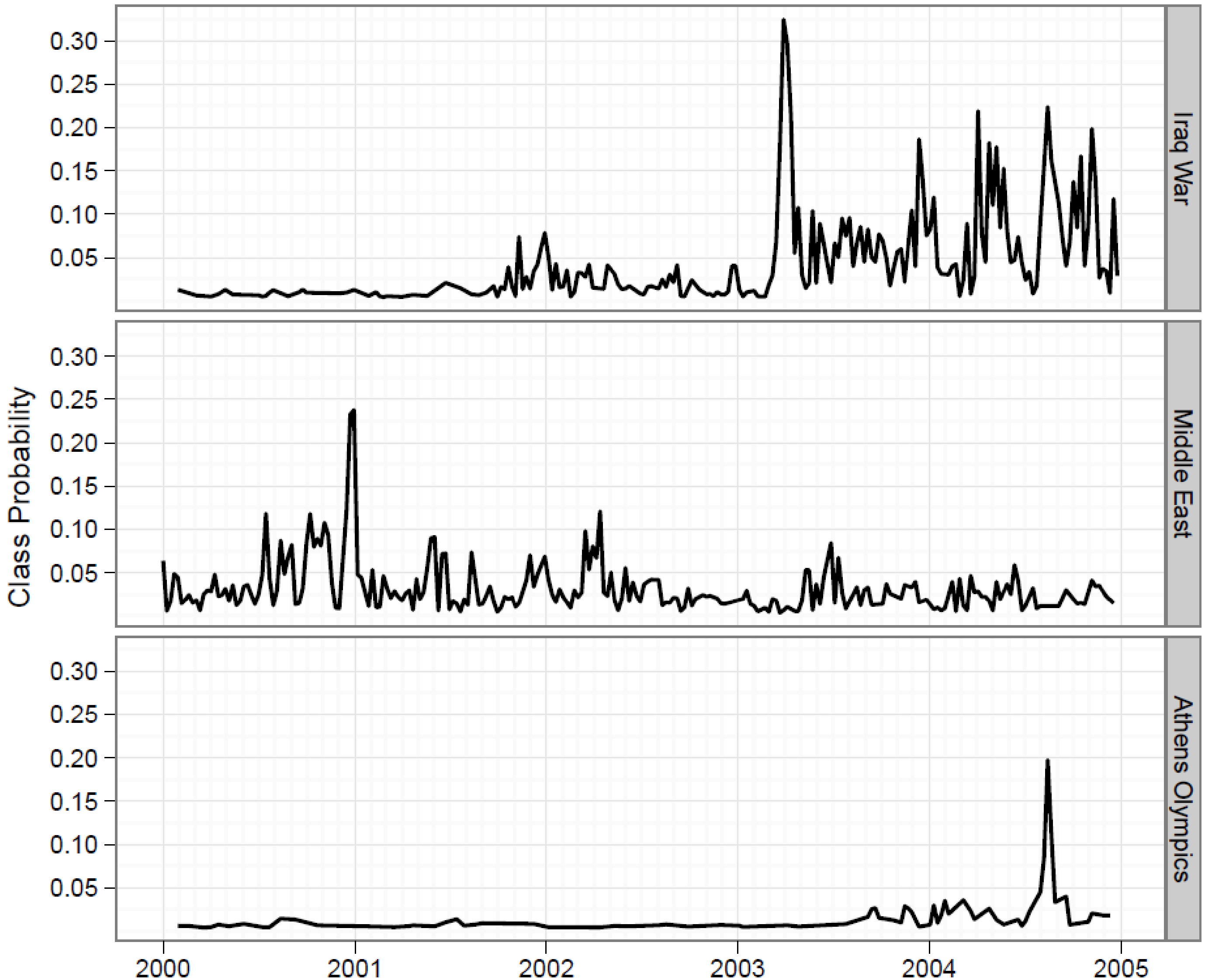


International Relations Data

(King, 2003)

40,000 events
2700 actors
171 action types

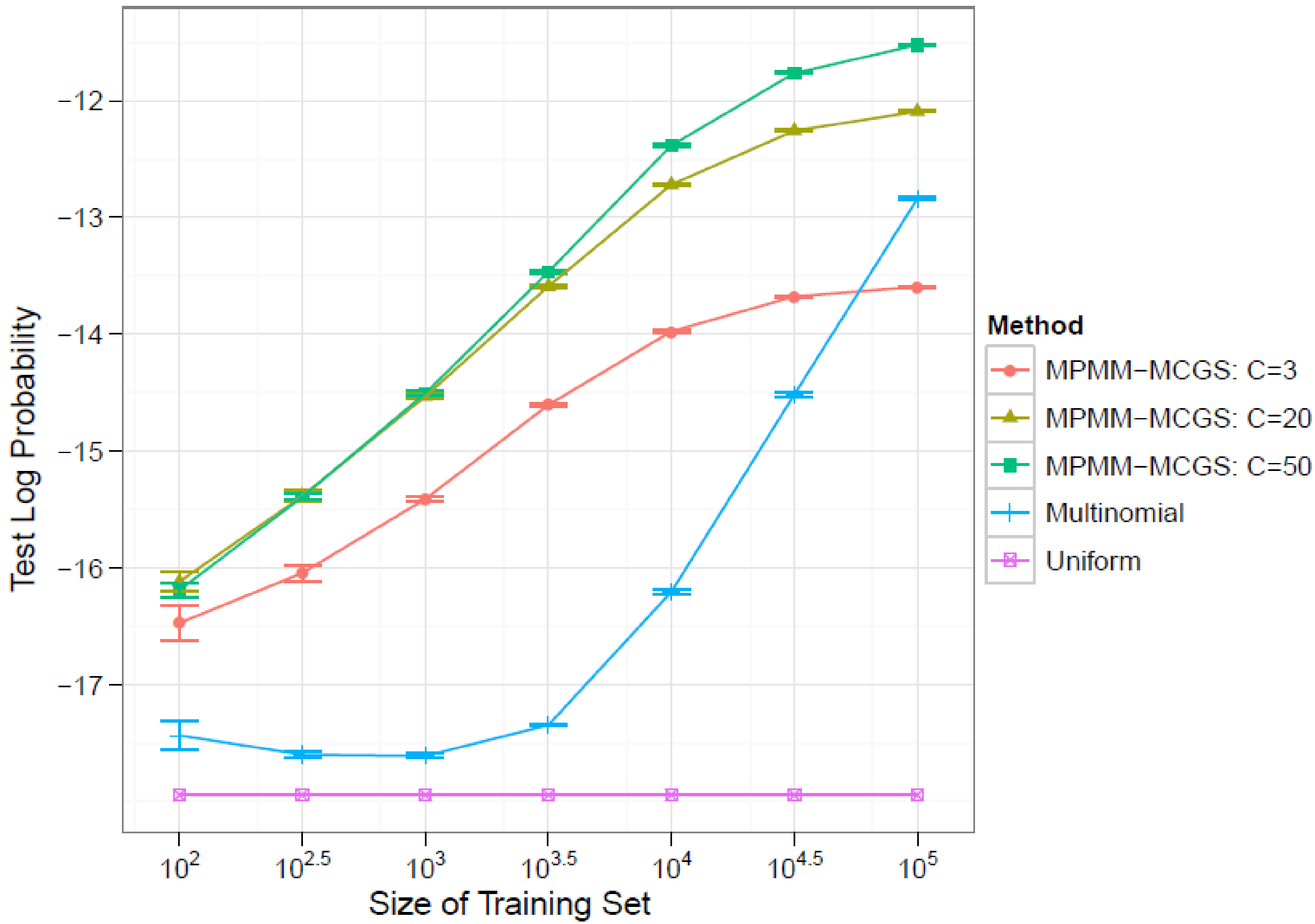
Top Senders	Pr.	Top Receivers	Pr.	Top Actions	Pr.
Class A					
United States : Government agents	0.47	Greece : NA	0.05	Sports contest	0.59
United States : Athletes	0.29	Australia : Government agents	0.02	Agree or accept	0.14
United States : Nominal agents	0.04	United Kingdom : NA	0.02	Optimistic comment	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

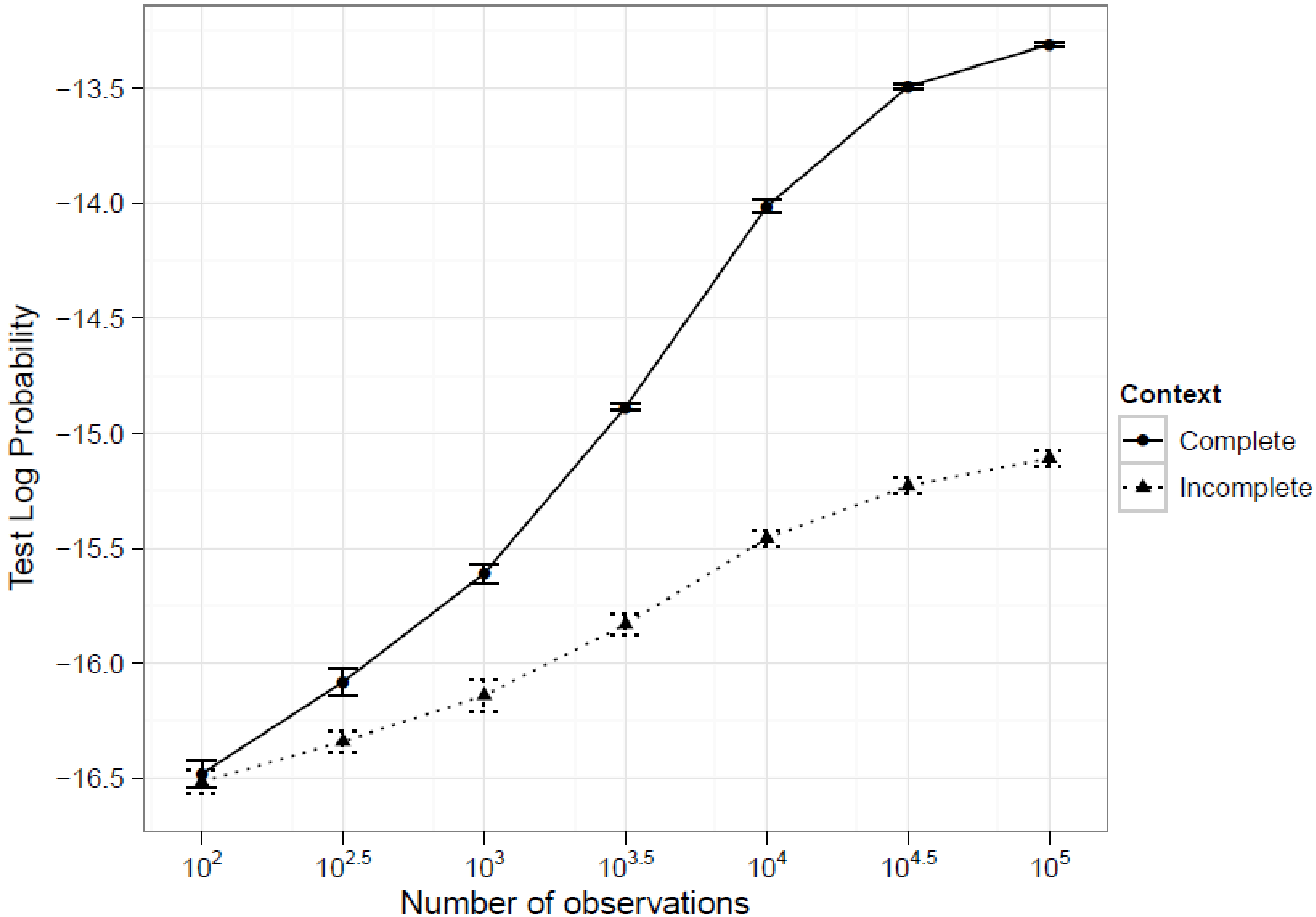


Class Probability
0.30
0.25
0.20
0.15
0.10
0.05
Iraq War
Middle East
Athens Olympics
2000
2001
2002
2003
2004
2005

Prediction and Evaluation

- **Use future data to evaluate predictive power and compare models**
- **Metrics**
  - Log score = log probability of events that actually occurred
  - Brier/MSE style scores
  - Ranking/ROC scores


Test Log Probability
−12
−13
−14
−15
−16
−17
10^2
10^2.5
10^3
10^3.5
10^4
10^4.5
10^5
Size of Training Set
Method
MPMM−MCGS: C=3
MPMM−MCGS: C=20
MPMM−MCGS: C=50
Multinomial
Uniform



Test Log Probability
−13.5
−14.0
−14.5
−15.0
−15.5
−16.0
−16.5
10^2
$10^{2.5}$
10^3
$10^{3.5}$
10^4
$10^{4.5}$
10^5
Number of observations
Context
Complete
Incomplete

Comments on Evaluation

- **Prediction on independent test data is critical**
  - Relatively easy to do with dynamic networks
  - Tricky to do with static networks (but see Hoff, 2009)

- **Caveat**
  - For link (or link probability) prediction it can be very difficult to beat relatively simple baselines, e.g.,
    - Graph(t+1) = Graph(t)
    - p(event) = smoothed estimate based on historical frequency of that pair

- **Solution?**
  - More interesting questions than just predicting what happens next, e.g
    - How likely is that group A will communicate with group B in the next k days?
    - If we have events with missing information, can we infer sender/receiver?
    - Can we detect significant shifts/non-stationarity?

What Next?

- Historically, social science applications of network analysis focused on understanding rather than prediction per se

- For data miners/computer scientists, predictive modeling plays a much more important role

- Key question: what are the important applications/problems that network/graph models can solve, that can't be solved by other means?
  - Candidates?
  - e.g., tools for egocentric modeling/analysis/management of personal communication data (email, social media, etc)
    - Change detection
    - Ranking of incoming communication events
    - ……

Summary

- **Latent variable models are useful for network modeling/prediction**
  - Broad “toolbox” of building blocks
  - May be scalable to large data sets
- **Latent models for dynamic data show promise**
- **Dynamic network data comes in multiple forms**
  - Aggregated/longitudinal data
  - Time-stamped event data

    (quite different in nature)
- **Models need to be evaluated via prediction on test data**

Resources

A survey of statistical network models

A. Goldenberg, A. Zheng, S. Fienberg, E. Airoldi, *Foundations and Trends in Machine Learning*, 2009

Multiplicative latent factor models for description and prediction of social networks

P. D. Hoff, *Computational and Mathematical Organization Theory , 2009.*

Random effects models for network data

P. D. Hoff, in *Dynamic Social Network Modeling and Analysis, 2003*

A relational event model for social action

C. E. Butts, *Sociological Methodology*, 2008

Slides from 2010 Whistler Summer School on Social Networks

http://people.cs.ubc.ca/~murphyk/pims2010Whistler/