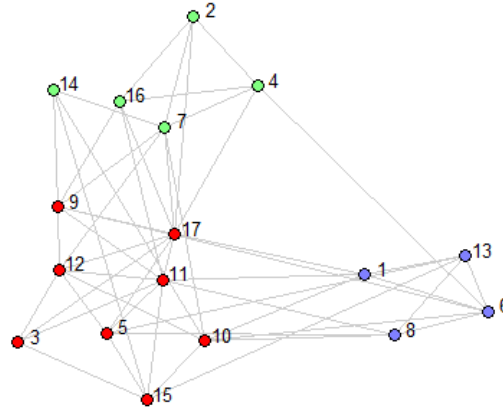




Generative models for social network data

Kevin S. Xu (University of Toledo)

James R. Foulds (University of California-San Diego)

SBP-BRiMS 2016 Tutorial

About Us

Kevin S. Xu

- Assistant professor at University of Toledo
- 3 years research experience in industry
- Research interests:
 - Machine learning
 - Network science
 - Physiological data analysis

James R. Foulds

- Postdoctoral scholar at UCSD
- Research interests:
 - Bayesian modeling
 - Social networks
 - Text
 - Latent variable models

Outline

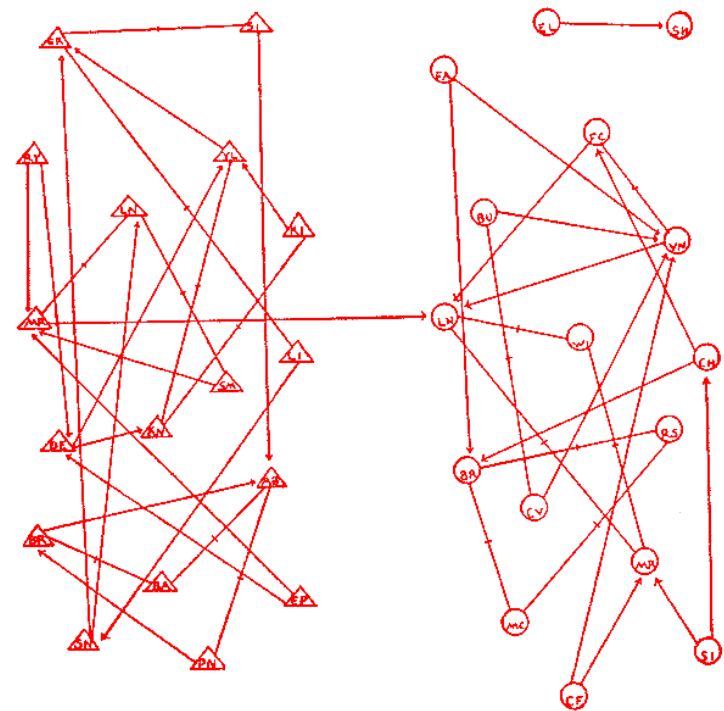
- Mathematical representations of social networks and generative models
 - Introduction to generative approach
 - Connections to sociological principles
- Fitting generative social network models to data
 - Example application scenarios
 - Model selection and evaluation
- Recent developments in generative social network models
 - Dynamic social network models

Social networks today



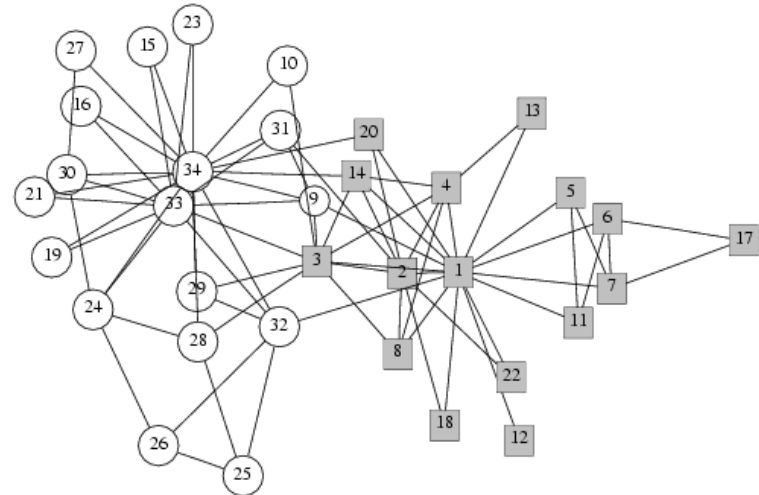
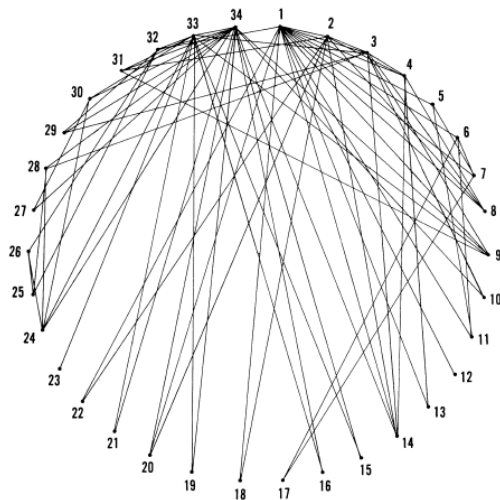
Social network analysis: an interdisciplinary endeavor

- Sociologists have been studying social networks for decades!
- First known empirical study of social networks: Jacob Moreno in 1930s
 - Moreno called them sociograms
- Recent interest in social network analysis (SNA) from physics, EECS, statistics, and many other disciplines



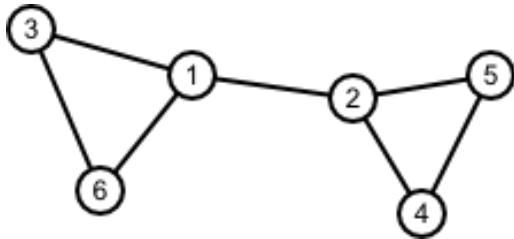
Social networks as graphs

- A social network can be represented by a **graph** $G = (V, E)$
 - V : vertices, **nodes**, or actors typically representing people
 - E : **edges**, links, or ties denoting relationships between nodes
 - Directed graphs used to represent asymmetric relationships
- Graphs have **no natural representation** in a geometric space
 - Two identical graphs drawn differently
 - **Moral: visualization provides very limited analysis ability**
 - How do we model and analyze social network data?

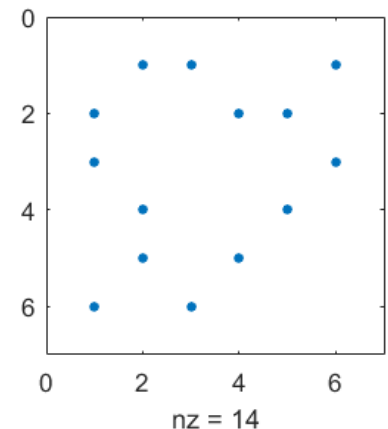


Matrix representation of social networks

- Represent graph by $n \times n$ **adjacency matrix** or sociomatrix $\mathbf{Y}$
 - $y_{ij} = 1$ if there is an edge between nodes i and j
 - $y_{ij} = 0$ otherwise



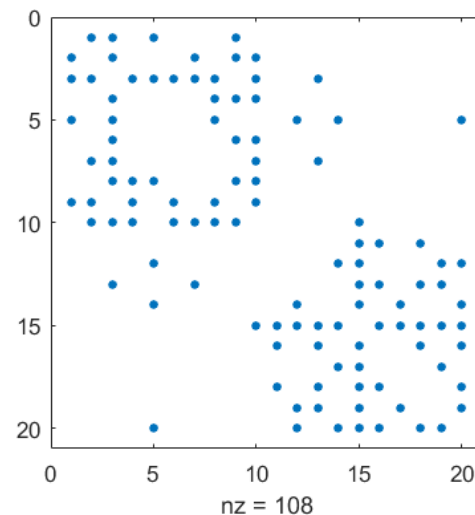
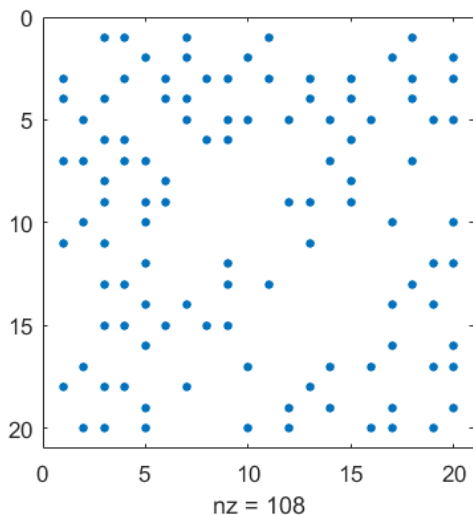
$$\mathbf{Y} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$



- Easily extended to directed and weighted graphs

Exchangeability of nodes

- Nodes are typically assumed to be (statistically) exchangeable by symmetry
- Row and column permutations to adjacency matrix do not change graph
 - Needs to be incorporated into social network models



Sociological principles related to edge formation

- **Homophily** or assortative mixing
 - Tendency for individuals to bond with similar others
 - Assortative mixing by age, gender, social class, organizational role, node degree, etc.
 - Results in **transitivity** (triangles) in social networks
 - “My friend of my friend is my friend”
- Equivalence of nodes
 - Two nodes are **structurally equivalent** if their relations to all other nodes are identical
 - Approximate equivalence recorded by similarity measure
 - Two nodes are **regularly equivalent** if their neighbors are similar (not necessarily common neighbors)

Brief history of social network models

- Early 1900s – sociology and social psychology precursors to SNA (Georg Simmel)
- 1930s – Graphical depictions of social networks: sociograms (Jacob Moreno)
- 1960s – Small world / 6-degrees of separation experiment (Stanley Milgram)
- 1970s – Mathematical models of social networks (Erdos-Renyi-Gilbert)
- 1980s – Statistical models (Holland and Leinhardt, Frank and Strauss)
- 1990s – Statistical physicists weigh in: preferential attachment, small world models, power-law degree distributions (Barabasi et al.)
- 2000s – Today – Machine learning approaches, latent variable models

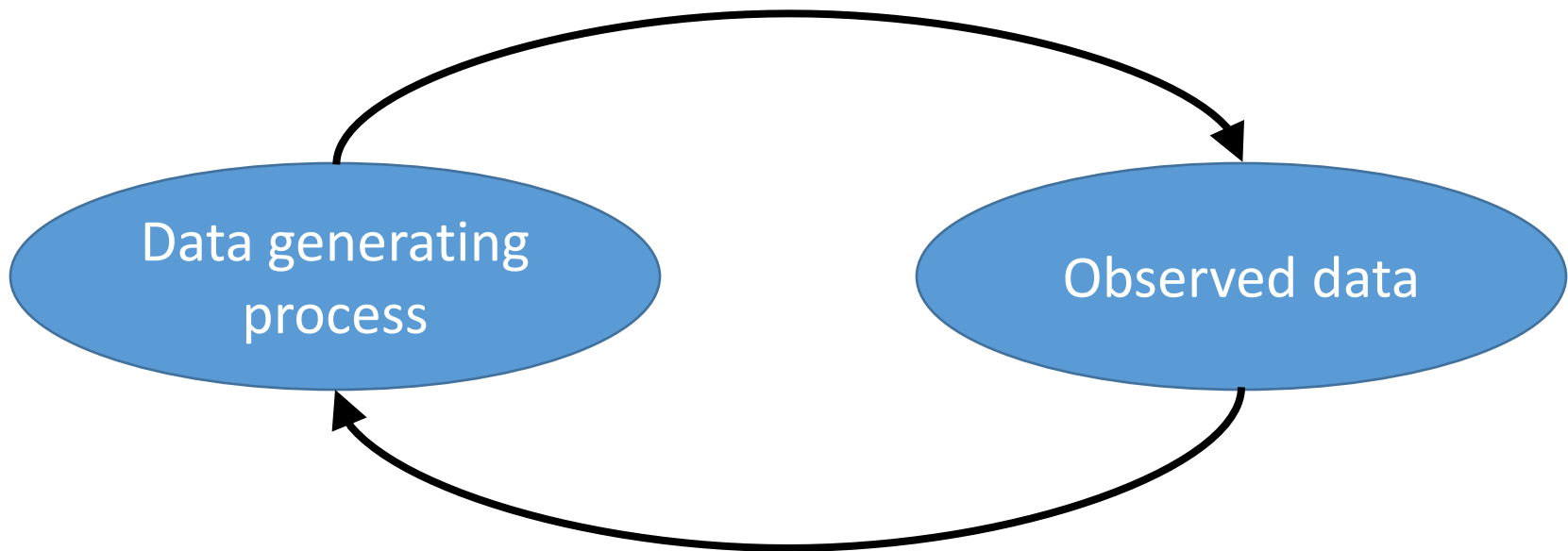
Generative models for social networks

- A **generative model** is one that can simulate new networks
- Two distinct schools of thought:
 - Probability models (non-statistical)
 - Typically simple, 1-2 parameters, not learned from data
 - Can be studied analytically
 - Statistical models
 - More parameters, latent variables
 - Learned from data via statistical techniques

Probability and Inference

Mathematics/physics: Erdős-Rényi, preferential attachment,...

Probability



Inference

Statistics/machine learning: ERGMs, latent variable models...

Probability models for networks (non-statistical)

Erdős-Rényi model

- There are two variants of this model
- The $G(N, E)$ model is a probability distribution over graphs with a **fixed number of edges**.
- It posits that all graphs on N nodes with E edges are equally likely

Erdős-Rényi model

- The $G(N, p)$ model posits that each edge is “on” with probability p
- Probability of adjacency matrix

$$Pr(\mathbf{Y}|p) = \prod_{i < j} p^{Y_{ij}} (1 - p)^{1 - Y_{ij}}$$

Erdős-Rényi model

- Adjacency matrix likelihood:

$$Pr(\mathbf{Y}|p) = \prod_{i < j} p^{Y_{ij}} (1 - p)^{1 - Y_{ij}}$$

- Number of edges is binomial.

$$Pr(E|p) = \binom{\binom{N}{2}}{E} p^E (1 - p)^{\binom{N}{2} - E}$$

- For large N, this is well approximated by a Poisson

$$E \approx \sim \text{Poisson}\left(\binom{N}{2} p\right)$$

Preferential attachment models

- The Erdős-Rényi model assumes nodes typically have about the same degree (# edges)
- Many real networks have a **degree distribution** following a power law (possibly controversial?)

$$P(k) \propto k^{-\lambda}$$

- **Preferential attachment** is a variant on the $G(N,p)$ model to address this (Barabasi and Albert, 1999)

Preferential attachment models

- Initially, no edges, and N_0 nodes.
- For each remaining node n
 - Add n to the network
 - For $i = 1:m$
 - Connect n to a random existing node with probability proportional to its degree (+ smoothing counts),

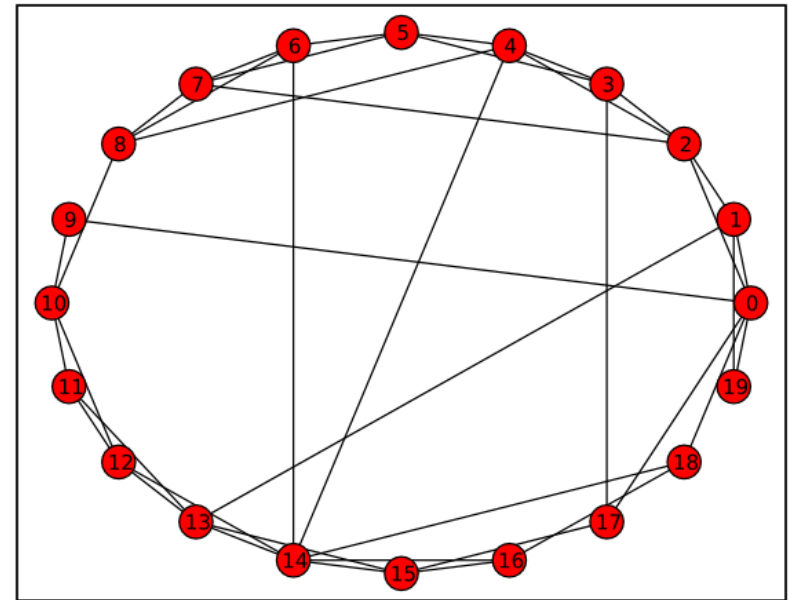
$$Pr(m) = \frac{k_m + k_0}{\sum_i (k_i + k_0)}$$

- A Polya urn process! Rich get richer.

Small world models (Watts and Strogatz)

- Start with nodes connected to K neighbors in a ring
- Randomly rewire each edge with probability β
- Has low average path length (**small world phenomenon**, “6-degrees of separation”)

Watts-Strogatz model $N=20, K=4, \beta=0.2$

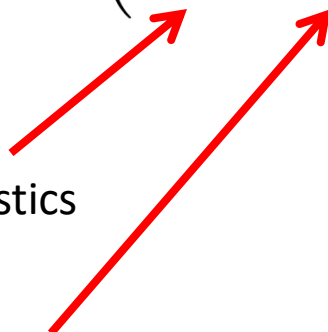


Statistical network models

Exponential family random graphs (ERGMs)

$$Pr(Y = y|\theta) = \frac{1}{Z(\theta)} \exp \left(\theta^\top S(y, \mathbf{X}) \right)$$

Arbitrary sufficient statistics



Covariates (gender, age, ...)

E.g. “how many males are friends with females”

Exponential family random graphs (ERGMs)

- Pros:
 - Powerful, flexible representation
 - Can encode complex theories, and do substantive social science
 - Handles covariates
 - Mature software tools available, e.g. ergm package for statnet

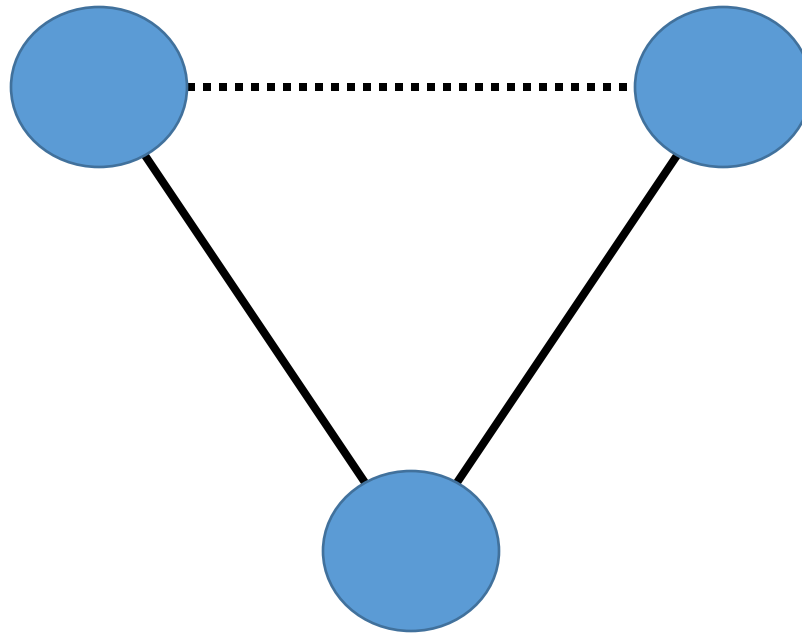
Exponential family random graphs (ERGMs)

- Cons:
 - Usual caveats of undirected models apply
 - Computationally intensive, especially learning
 - Inference may be intractable, due to partition function

Exponential family random graphs (ERGMs)

- Cons:
 - Usual caveats of undirected models apply
 - Computationally intensive, especially learning
 - Inference may be intractable, due to partition function
 - **Model degeneracy** can easily happen
 - *“a seemingly reasonable model can actually be such a bad mis-specification for an observed dataset as to render the observed data virtually impossible”*
 - Goodreau (2007)

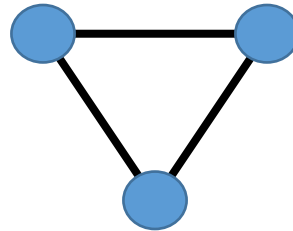
Triadic closure



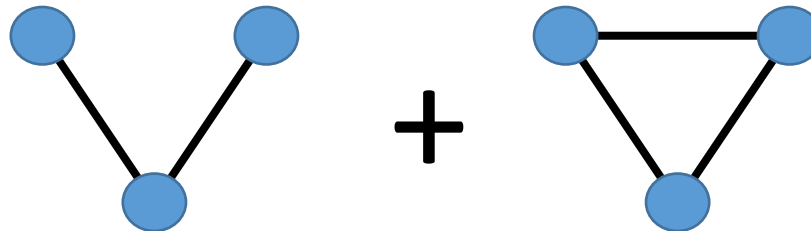
If two people have a friend in common, then there is an increased likelihood that they will become friends themselves at some point in the future.

Measuring triadic closure

- Mean clustering co-efficient:



$$C = \frac{\# \text{ triads with all three edges}}{\# \text{ triads with at least two edges}}$$



Simple ERGM for triadic closure
leads to model degeneracy

$$Pr(Y = y|\theta) = \frac{1}{Z(\theta)} \exp \left(\theta^\top S(y, \mathbf{X}) \right)$$

$$\theta = [\text{edge density, mean clustering coefficient}]^\top$$

Simple ERGM for triadic closure leads to model degeneracy

$$Pr(Y = y|\theta) = \frac{1}{Z(\theta)} \exp \left(\theta^\top S(y, \mathbf{X}) \right)$$

$$\theta = [\text{edge density, mean clustering coefficient}]^\top$$

Depending on parameters, we could get:

- Graph is empty with probability close to 1
- Graph is full with probability close to 1
- Density, clustering distribution is bimodal, with little mass on desired density and triad closure



MLE may not exist!

Distribution of Graphs from this model

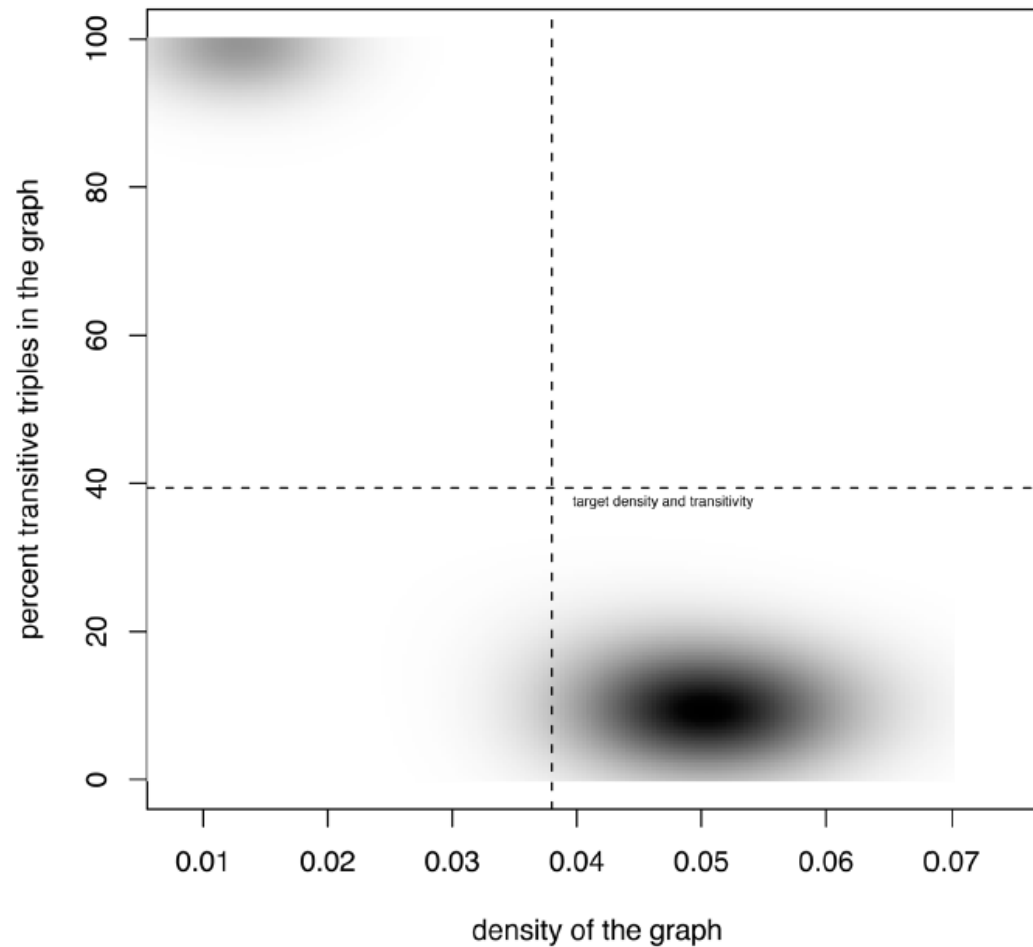
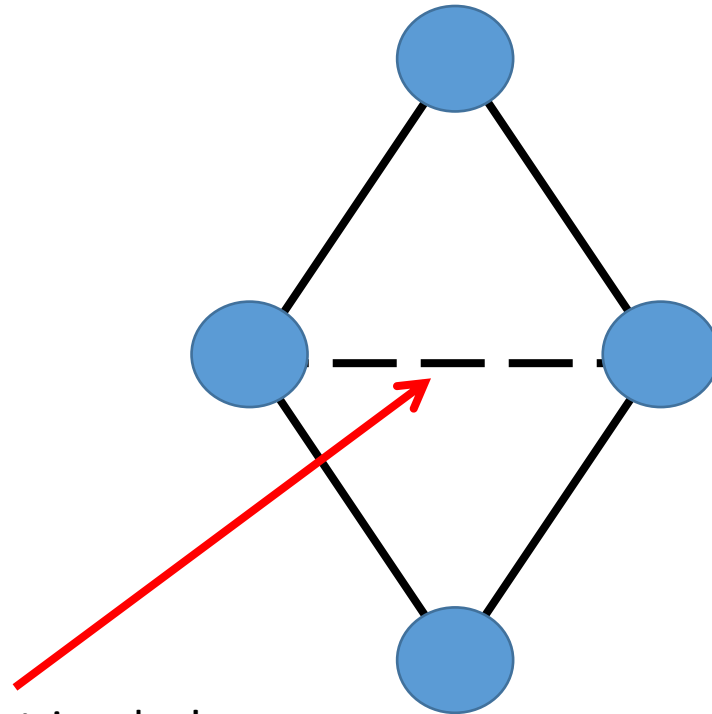


Figure 1.

Darker gray indicates higher probability density in this plot showing the true distribution of networks according to a particular two-statistic ERGM containing edge density and mean clustering coefficient. The population mean vector, specified by a particular choice of the model parameters, is shown at the intersection of the two dotted lines. The fact that there is very little probability mass near this mean is emblematic of degeneracy.

What is the problem?



Completes two triangles!

If an edge completes more triangles, it becomes overwhelming likely to exist.
This propagates to create more triangles ...

Solution

- Change the model so that there are diminishing returns for completing more triangles
 - A different natural parameter for each possible number of triangles completed by one edge
 - Natural parameters $\eta(\theta)$ parameterized by a lower-dimensional θ , e.g. encoding geometrically decreasing weights (**curved exponential family**)
- **Moral of the story:** ERGMS are powerful, but require care and expertise to perform well

Latent variable models for social networks

- Model where observed variables are dependent on a set of unobserved or **latent** variables
 - Observed variables assumed to be conditionally independent given latent variables
- Why latent variable models?
 - Adjacency matrix $\mathbf{Y}$ is invariant to row and column permutations
 - Aldous-Hoover theorem implies existence of a latent variable model of form

$$y_{ij} = h(\theta, z_i, z_j, \epsilon_{ij})$$

for iid latent variables z_i and some function h

Latent variable models for social networks

- Latent variable models allow for heterogeneity of nodes in social networks
 - Each node (actor) has a latent variable $\mathbf{z}_i$
 - Probability of forming edge between two nodes is **independent** of all other node pairs given values of latent variables

$$p(\mathbf{Y}|\mathbf{Z}, \theta) = \prod_{i \neq j} p(y_{ij} | \mathbf{z}_i, \mathbf{z}_j, \theta)$$

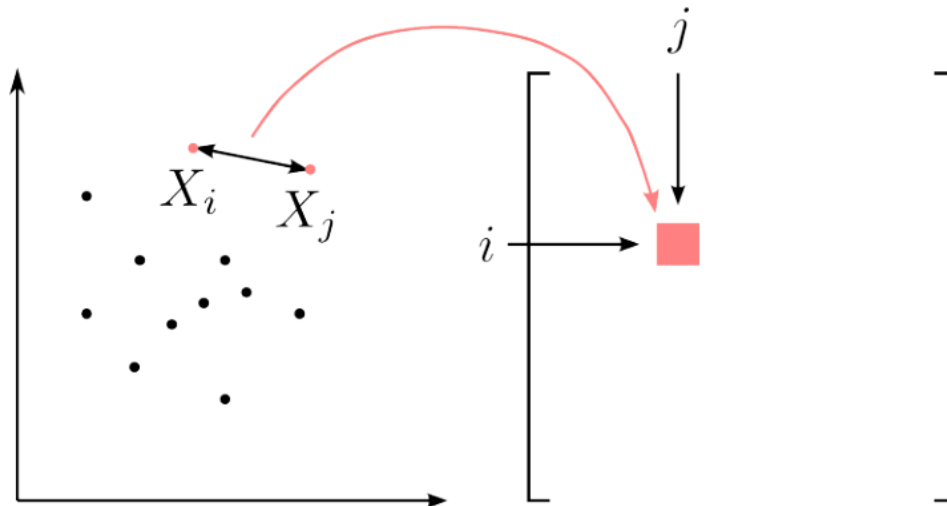
- Ideally latent variables should provide an **interpretable** representation

(Continuous) latent space model

- Motivation: homophily or assortative mixing
 - Probability of edge between two nodes increases as characteristics of the nodes become more similar
- Represent nodes in an unobserved (latent) space of characteristics or “social space”
- Small distance between 2 nodes in latent space → high probability of edge between nodes
 - Induces transitivity: observation of edges (i, j) and (j, k) suggests that i and k are not too far apart in latent space → more likely to also have an edge

(Continuous) latent space model

- (Continuous) latent space model (LSM) proposed by Hoff et al. (2002)
 - Each node has a latent position $\mathbf{z}_i \in \mathbb{R}^d$
 - Probabilities of forming edges depend on **distances** between latent positions
 - Define pairwise **affinities** $\psi_{ij} = \theta - \|\mathbf{z}_i - \mathbf{z}_j\|_2$



Latent space model: generative process

1. Sample node positions in latent space

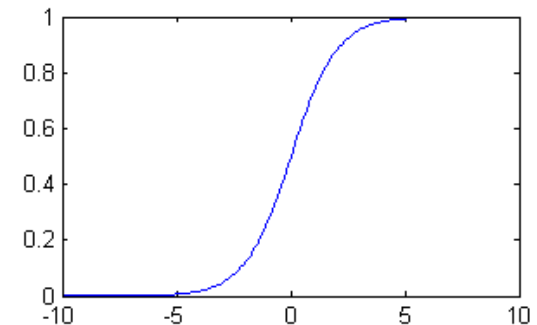
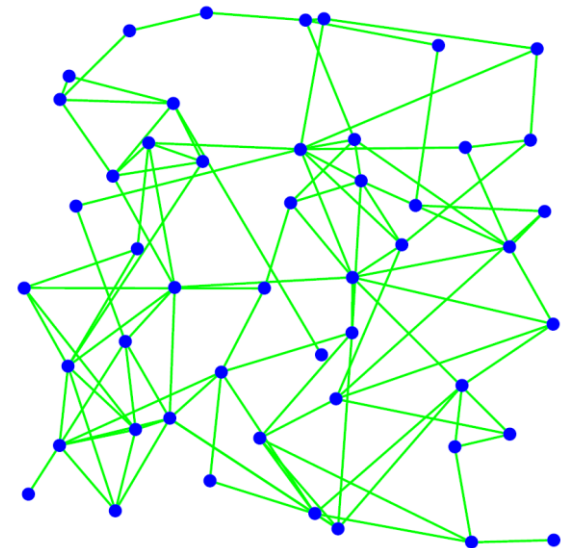
$$\mathbf{z}_i \sim \text{Gaussian}(\mathbf{0}, \kappa \mathbf{I})$$

2. Compute affinities between all pairs of nodes

$$\psi_{ij} = \theta - \|\mathbf{z}_i - \mathbf{z}_j\|_2$$

3. Sample edges between all pairs of nodes

$$P(Y_{ij} = 1 | \psi_{ij}) = \sigma(\psi_{ij})$$



Advantages and disadvantages of latent space model

- Advantages of latent space model
 - Visual and interpretable spatial representation of network
 - Models homophily (assortative mixing) well via transitivity
- Disadvantages of latent space model
 - 2-D latent space representation often may not offer enough degrees of freedom
 - Cannot model disassortative mixing (people preferring to associate with people with different characteristics)

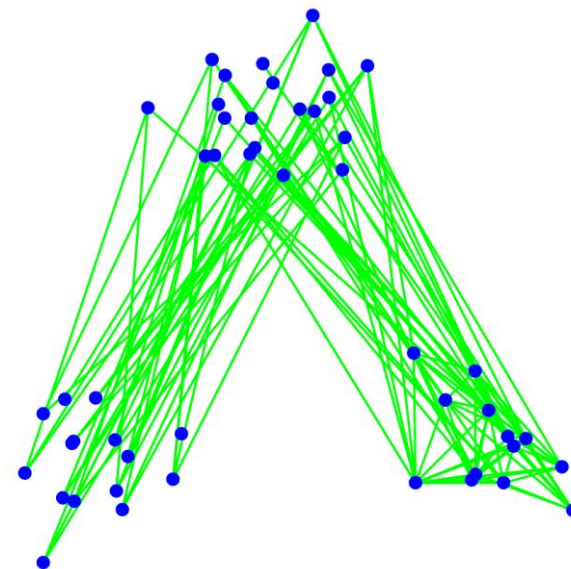
Stochastic block model (SBM)

- First formalized by Holland et al. (1983)
- Also known as multi-class Erdős-Rényi model
- Each node has categorical latent variable $z_i \in \{1, \dots, K\}$ denoting its class or group
- Probabilities of forming edges depend on class memberships of nodes ($K \times K$ matrix W)
 - Groups often interpreted as functional roles in social networks

$$Y \sim \begin{array}{|c|c|c|} \hline \text{red box } W_{11} & \cdots & \text{yellow box } W_{1K} \\ \hline \vdots & \ddots & \vdots \\ \hline \text{blue box } W_{K1} & \cdots & \text{green box } W_{KK} \\ \hline \end{array}$$

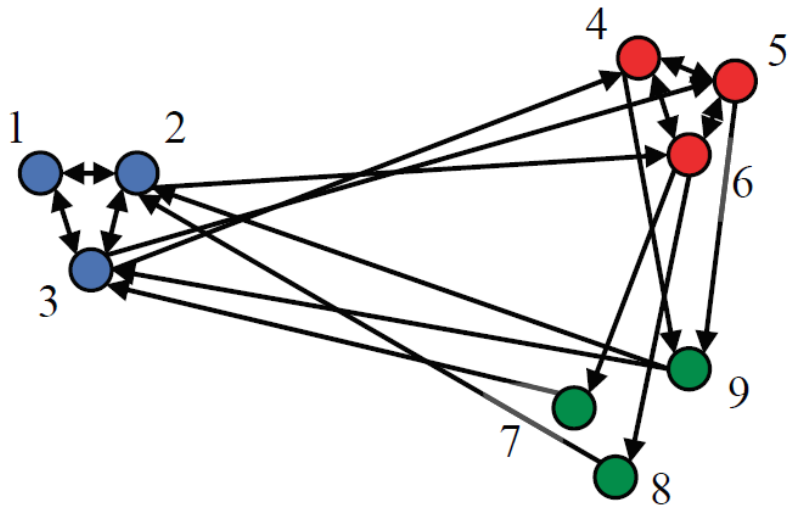
Stochastic equivalence and block models

- Stochastic equivalence: generalization of structural equivalence
- Group members have identical **probabilities** of forming edges to members other groups
 - Can model both assortative and disassortative mixing



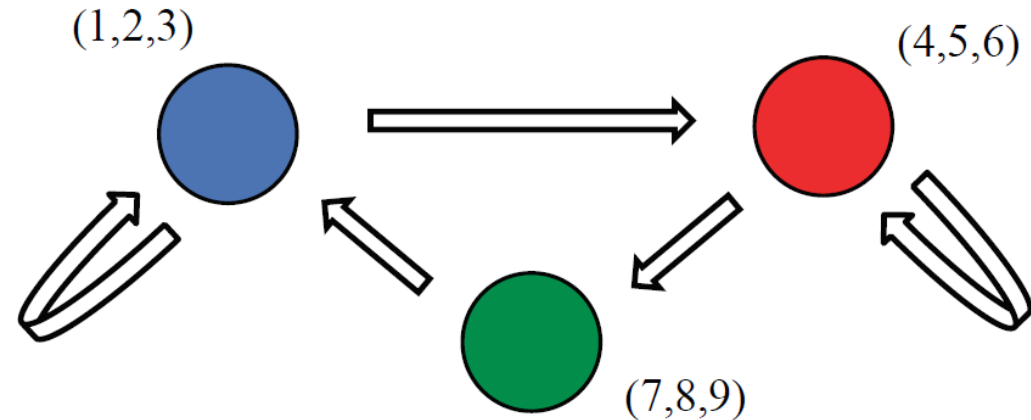
Stochastic equivalence vs community detection

Original graph



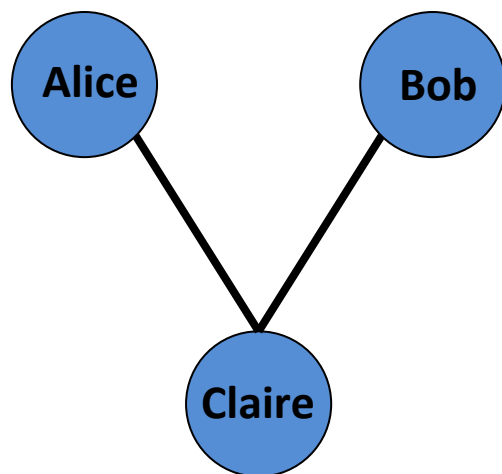
Stochastically equivalent, but
are not densely connected

Blockmodel



Stochastic blockmodel

Latent representation

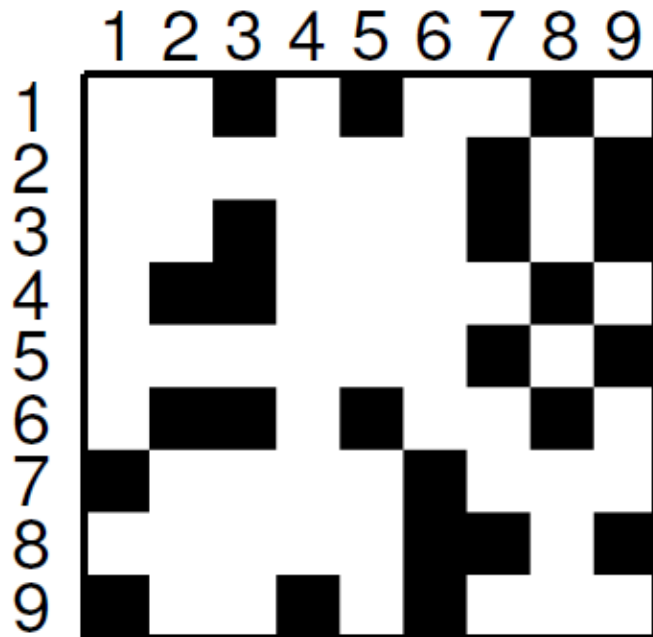


$\mathbf{Z} =$

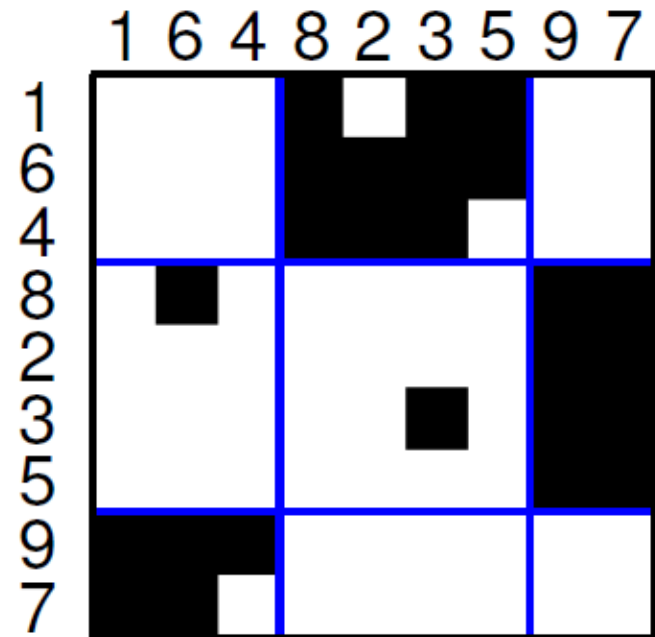
	UCSD	UCI	UCLA
Alice	1		
Bob			1
Claire		1	

Reordering the matrix to show the inferred block structure

Input

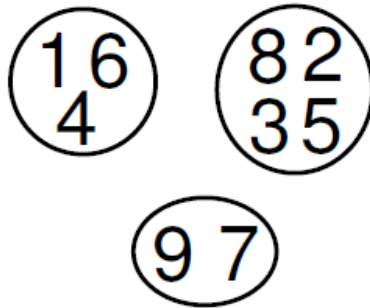


Output



Model structure

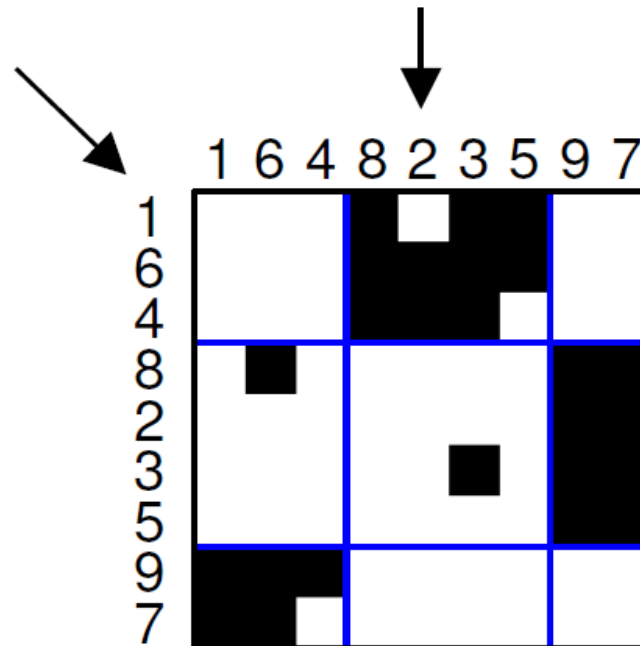
Latent groups Z



0.1	0.9	0.1
0.1	0.1	0.9
0.9	0.1	0.1

Interaction matrix W

(probability of an edge from block k to block k')



Stochastic block model generative process

$W_{kk'}$: Probability that a node in group k connects to a node in k'

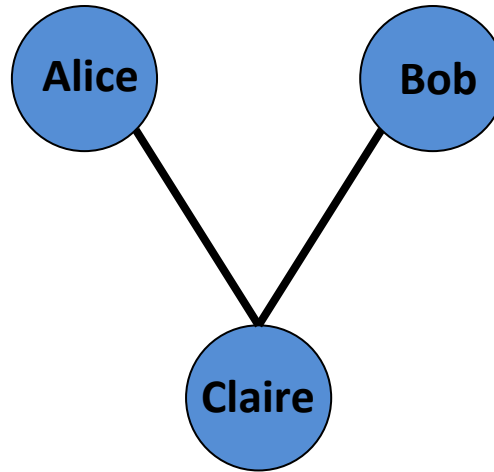
z_i : Latent group assignment for node i

For each pair of nodes (i, j)

$$Y_{ij} \sim \text{Bernoulli}(W_{z_i, z_j})$$

Stochastic block model

Latent representation



Nodes assigned to only one latent group.

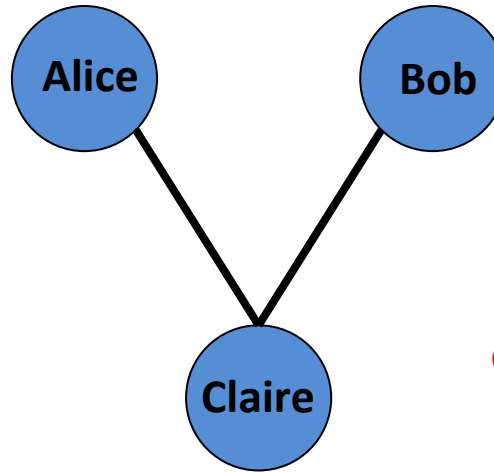
Not always an appropriate assumption



$\mathbf{Z} =$

	Running	Dancing	Fishing
Alice	1		
Bob			1
Claire		1	

Mixed membership stochastic blockmodel (MMSB)



Nodes represented by **distributions**
over latent groups (roles)

$\mathbf{Z} =$

	Running	Dancing	Fishing
Alice	0.4	0.4	0.2
Bob	0.5	0.5	
Claire	0.1		0.9

Mixed membership stochastic blockmodel (MMSB)

$\pi^{(i)}$: Mixed membership vector for node i

$W_{kk'}$: Probability that group k connects to group k'

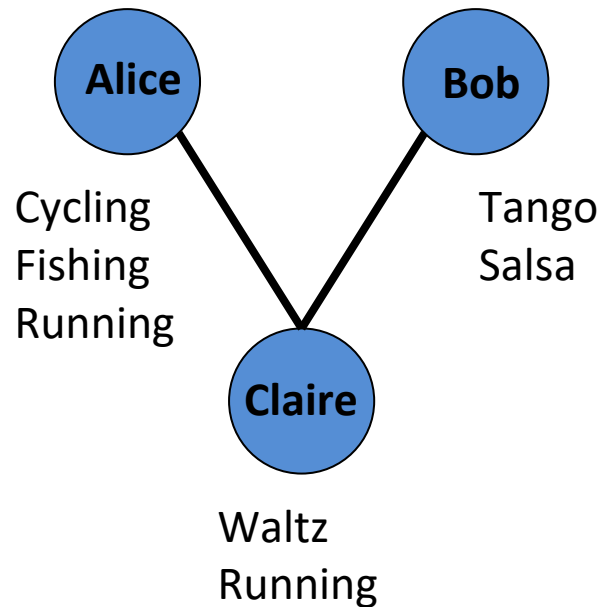
For each pair of nodes (i, j)

$$z_i^{(ij)} \sim \text{discrete}(\pi^{(i)})$$

$$z_j^{(ij)} \sim \text{discrete}(\pi^{(j)})$$

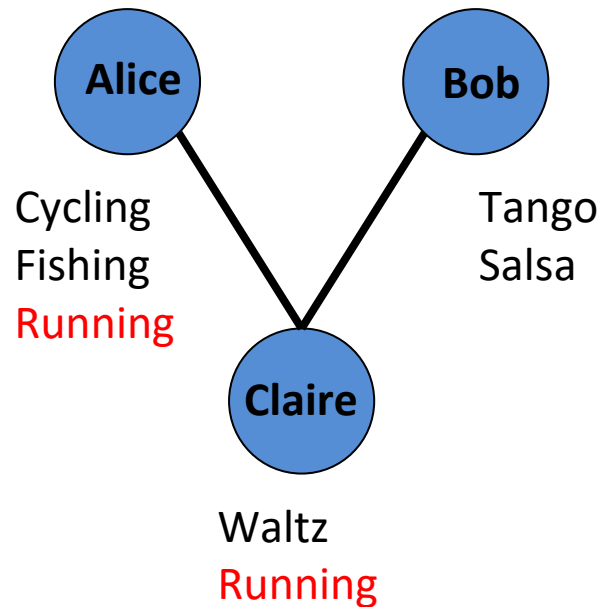
$$Y_{ij} \sim \text{Bernoulli}(W_{z_i^{(ij)}, z_j^{(ij)}})$$

Latent feature models



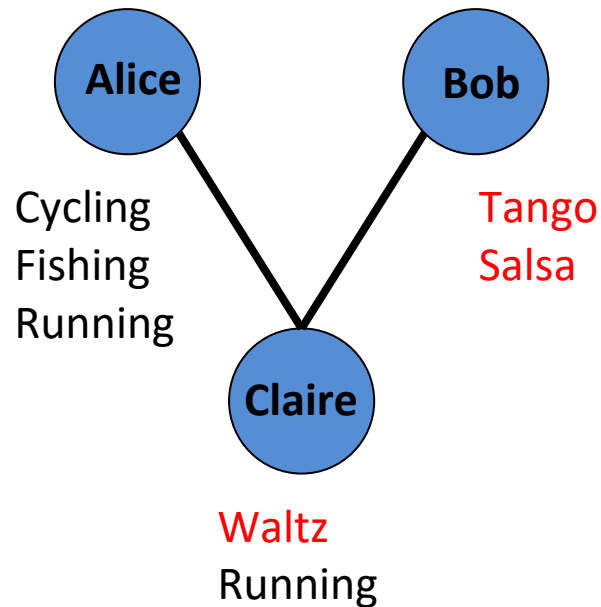
Mixed membership implies a kind of “conservation of (probability) mass” constraint:
If you like cycling more, you must like running less, to sum to one

Latent feature models



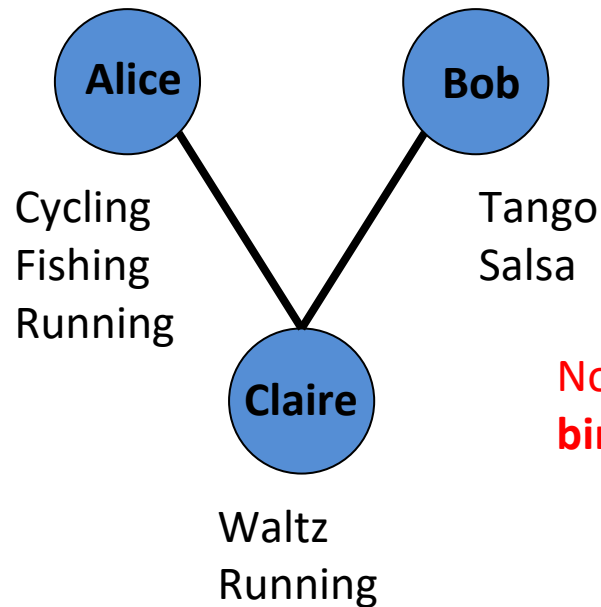
Mixed membership implies a kind of “conservation of (probability) mass” constraint:
If you like cycling more, you must like running less, to sum to one

Latent feature models



Mixed membership implies a kind of “conservation of (probability) mass” constraint:
If you like cycling more, you must like running less, to sum to one

Latent feature models



Nodes represented by
binary vector of latent features

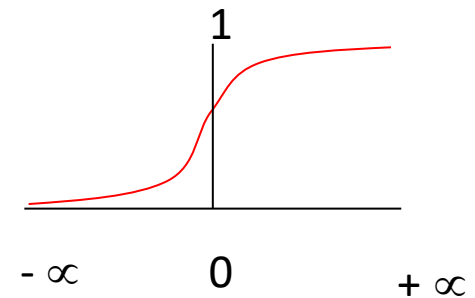
$Z =$

	Cycling	Fishing	Running	Tango	Salsa	Waltz
Alice						
Bob						
Claire						

Latent feature models

- Latent Feature Relational Model LFRM
(Miller, Griffiths, Jordan, 2009) likelihood model:

$$P(Y_{ij} = 1 | \dots) = \sigma(\mathbf{z}_i \mathbf{W} \mathbf{z}_j^\top)$$



- “If I have feature k , and you have feature l , add W_{kl} to the log-odds of the probability we interact”
- Can include terms for network density, covariates, popularity,..., as in the p2 model

Outline

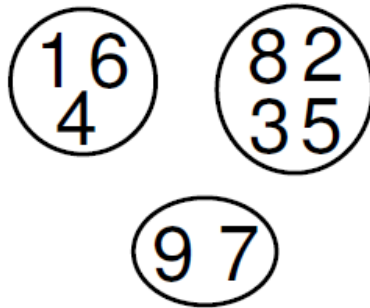
- Mathematical representations of social networks and generative models
 - Introduction to generative approach
 - Connections to sociological principles
- Fitting generative social network models to data
 - Example application scenarios
 - Model selection and evaluation
- Recent developments in generative social network models
 - Dynamic social network models

Application 1: Facebook wall posts

- Network of wall posts on Facebook collected by Viswanath et al. (2009)
 - Nodes: Facebook users
 - Edges: directed edge from i to j if i posts on j 's Facebook wall
- What model should we use?
 - (Continuous) latent space and latent feature models do not handle directed graphs in a straightforward manner
 - Wall posts might not be transitive, unlike friendships
- Stochastic block model might not be a bad choice as a starting point

Model structure

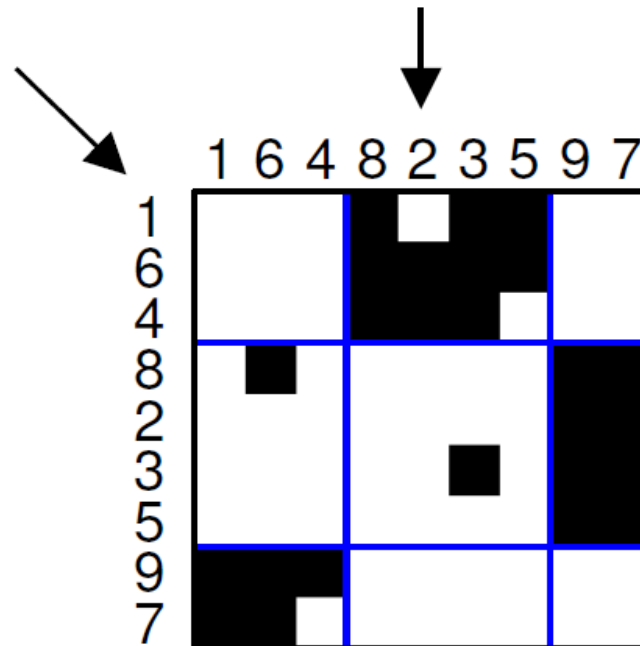
Latent groups Z



0.1	0.9	0.1
0.1	0.1	0.9
0.9	0.1	0.1

Interaction matrix W

(probability of an edge from block k to block k')



Fitting stochastic block model

- A priori block model: assume that class (role) of each node is given by some other variable
 - Only need to estimate $W_{kk'}$: probability that node in class k connects to node in class k' for all k, k'

- Likelihood given by

$$Pr(\mathbf{Y}|\mathbf{W}, \mathbf{Z}) = \exp \left\{ \sum_{k=1}^K \sum_{k'=1}^K [m_{kk'} \log W_{kk'} + (n_{kk'} - m_{kk'}) \log(1 - W_{kk'})] \right\}$$

Number of actual edges in block (k, k')

Number of possible edges in block (k, k')

- Maximum-likelihood estimate (MLE) given by

$$\hat{W}_{kk'} = \frac{m_{kk'}}{n_{kk'}}$$

Estimating latent classes

- Latent classes (roles) are unknown in this data set
 - First estimate latent classes $\mathbf{Z}$ then use MLE for $\mathbf{W}$
- MLE over latent classes is intractable!
 - $\sim K^N$ possible latent class vectors
- Spectral clustering techniques have been shown to accurately estimate latent classes
 - Use singular vectors of (possibly transformed) adjacency matrix to estimate classes
 - Many variants with differing theoretical guarantees

Spectral clustering for directed SBMs

1. Compute singular value decomposition
 $Y = U\Sigma V^T$
2. Retain only first K columns of U, V and first K rows and columns of Σ
3. Define coordinate-scaled singular vector matrix
 $\tilde{Z} = [U\Sigma^{1/2} \ V\Sigma^{1/2}]$
4. Run k-means clustering on rows of $\tilde{Z}$ to return estimate $\hat{Z}$ of latent classes

Scales to networks with thousands of nodes!

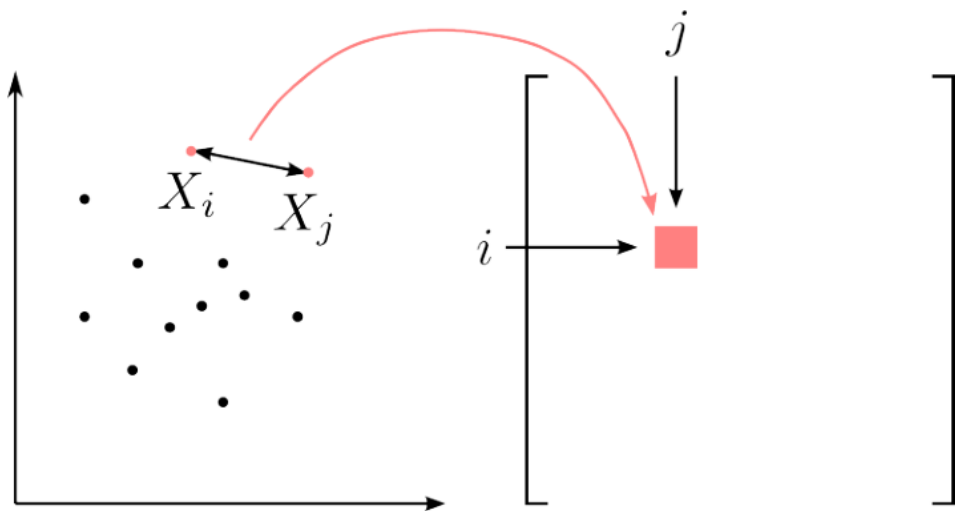
Demo of SBM on Facebook wall post network

Application 2: social network of bottlenose dolphin interactions

- Data collected by marine biologists observing interactions between 62 bottlenose dolphins
 - Introduced to network science community by Lusseau and Newman (2004)
 - Nodes: dolphins
 - Edges: undirected relations denoting frequent interactions between dolphins
- What model should we use?
 - Social interactions here are in a group setting so lots of transitivity may be expected
 - Interactions associated by physical proximity
 - Use latent space model to estimate latent positions

(Continuous) latent space model

- (Continuous) latent space model (LSM) proposed by Hoff et al. (2002)
 - Each node has a latent position $\mathbf{z}_i \in \mathbb{R}^d$
 - Probabilities of forming edges depend on **distances** between latent positions
 - Define pairwise **affinities** $\psi_{ij} = \theta - \|\mathbf{z}_i - \mathbf{z}_j\|_2$



$$p(Y|Z, \theta) = \prod_{i \neq j} \frac{e^{y_{ij}\psi_{ij}}}{1 + e^{\psi_{ij}}}$$

Estimation for latent space model

- Maximum-likelihood estimation
 - Log-likelihood is concave in terms of pairwise distance matrix D but not in latent positions Z
 - First find MLE in terms of D then use multi-dimensional scaling (MDS) to get initialization for Z
 - Faster approach: replace D with shortest path distances in graph then use MDS
 - Use non-linear optimization to find MLE for Z
- Latent space dimension often set to 2 to allow visualization using scatter plot

Scales to ~1000 nodes

Demo of latent space model on dolphin network

Bayesian inference

- As a Bayesian, all you have to do is write down your prior beliefs, write down your likelihood, and apply Bayes ' rule,

$$Pr(\theta|\mathbf{X}) = \frac{Pr(\mathbf{X}|\theta)Pr(\theta)}{Pr(\mathbf{X})}$$

Elements of Bayesian Inference

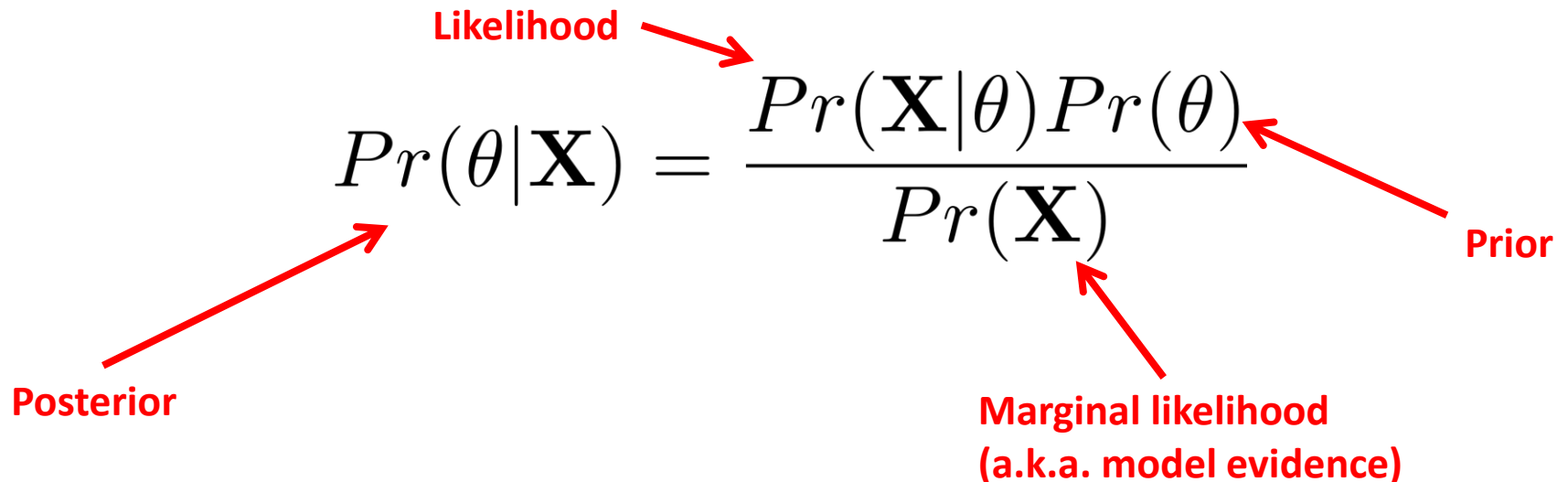


Diagram illustrating the components of Bayesian Inference:

$$Pr(\theta|\mathbf{X}) = \frac{Pr(\mathbf{X}|\theta)Pr(\theta)}{Pr(\mathbf{X})}$$

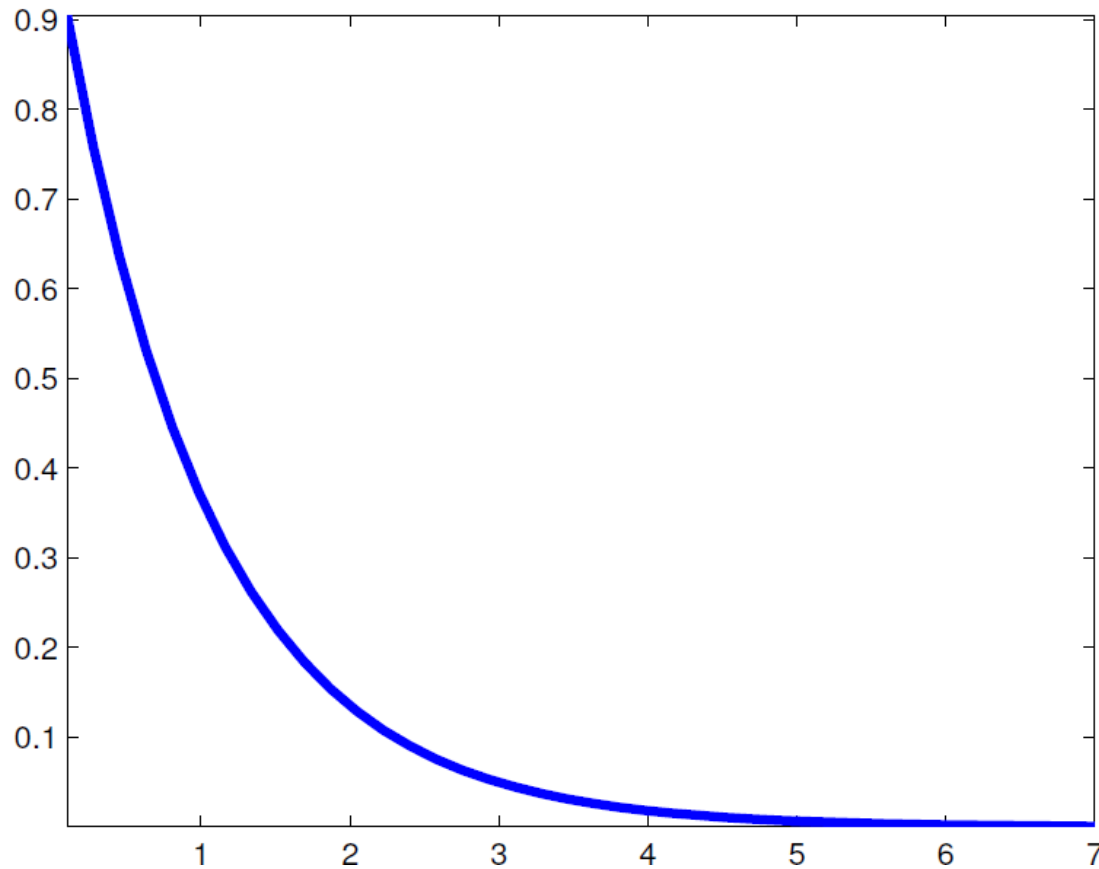
The components are labeled with red arrows:

- Posterior** points to $Pr(\theta|\mathbf{X})$.
- Likelihood** points to $Pr(\mathbf{X}|\theta)$.
- Prior** points to $Pr(\theta)$.
- Marginal likelihood (a.k.a. model evidence)** points to $Pr(\mathbf{X})$.

$$Pr(\mathbf{X}) = \int Pr(\mathbf{X}|\theta)Pr(\theta)d\theta$$

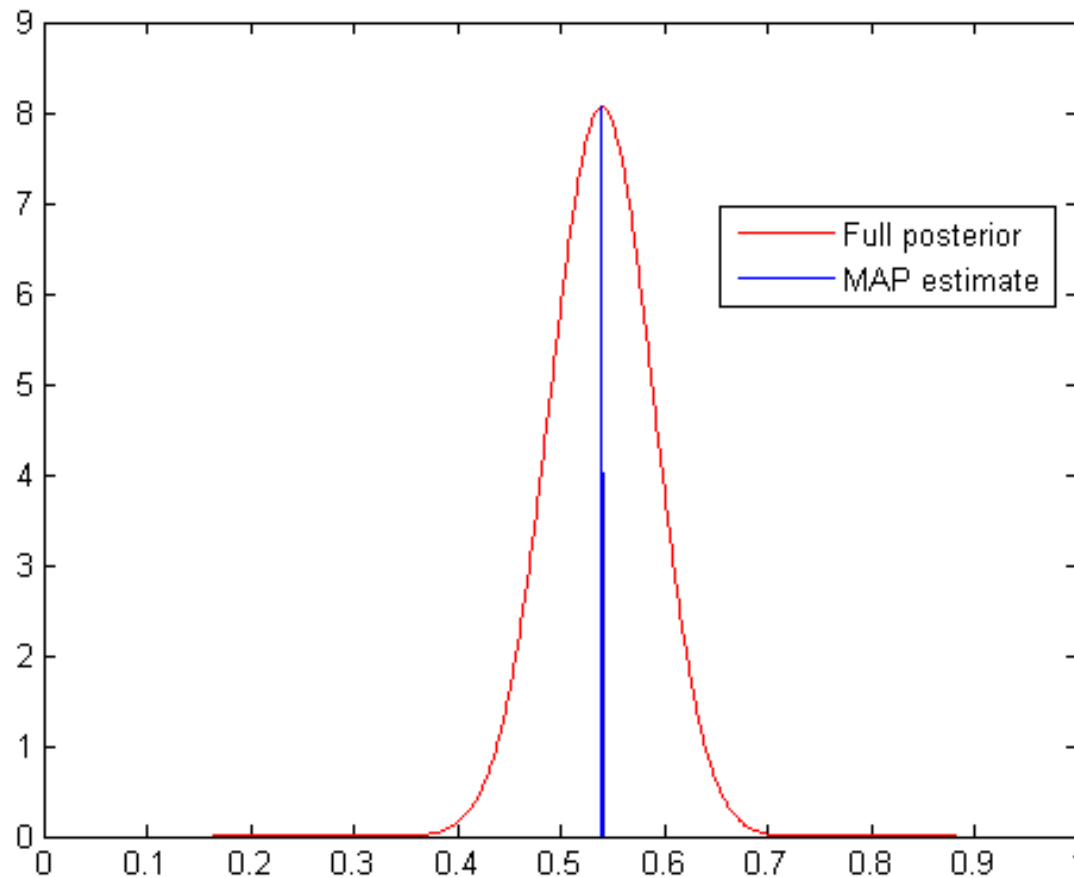
is a normalization constant that does not depend on the value of θ . It is the probability of the data under the model, marginalizing over all possible θ 's.

The full posterior distribution can be very useful



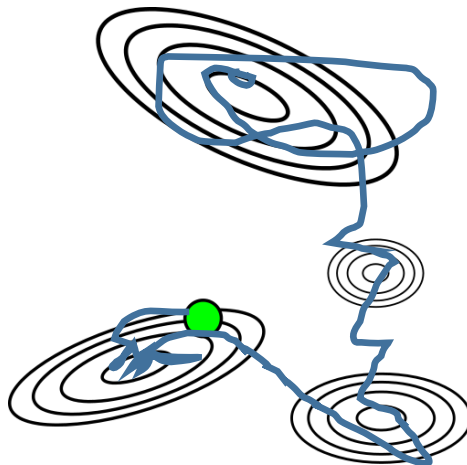
The mode (MAP estimate) is unrepresentative of the distribution

MAP estimate can result in overfitting



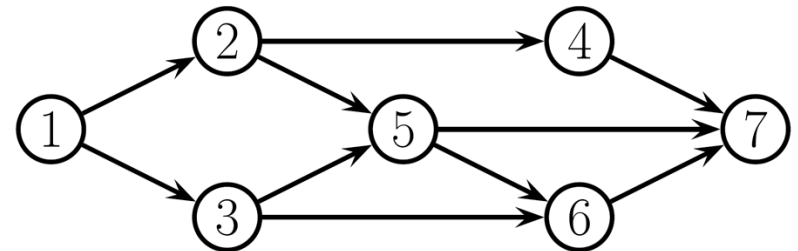
Markov chain Monte Carlo

- **Goal:** approximate/summarize a distribution, e.g. the posterior, with a set of samples
- **Idea:** use a Markov chain to simulate the distribution and draw samples



Gibbs sampling

- Sampling from a complicated distribution, such as a Bayesian posterior, can be hard.
- Often, sampling one variable at a time, given all the others, is much easier.



- Graphical models:
Graph structure gives us Markov blanket

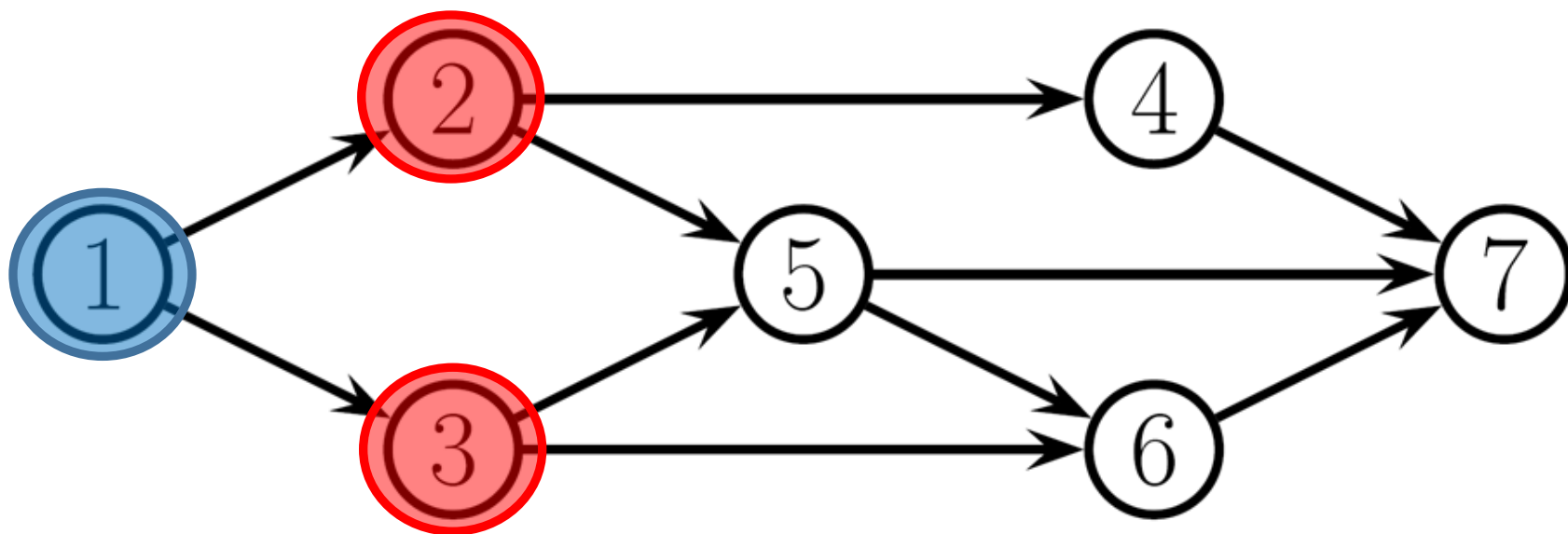
Gibbs sampling

- Update variables one at a time by drawing from their conditional distributions

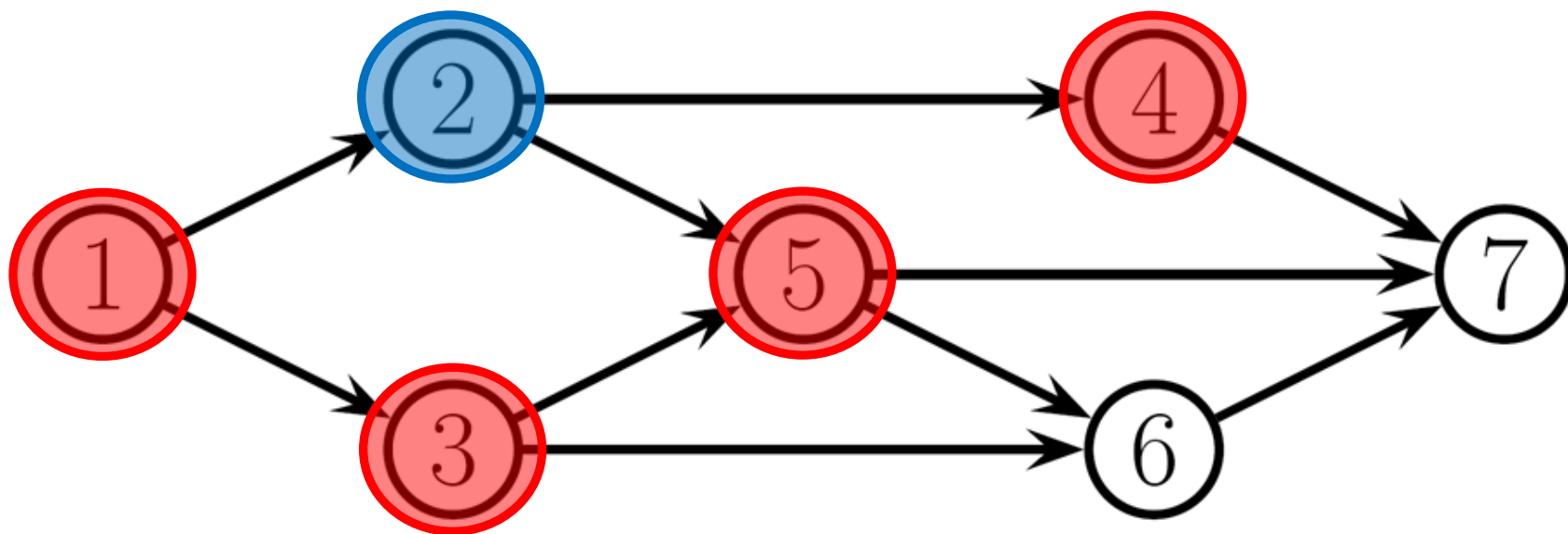
$$\mathbf{z}_i := \mathbf{z}_i^{(new)}, \mathbf{z}_i^{(new)} \sim Pr(\mathbf{z}_i | \mathbf{z}_{\neg i})$$

- In each iteration, sweep through and update all of the variables, in any order.

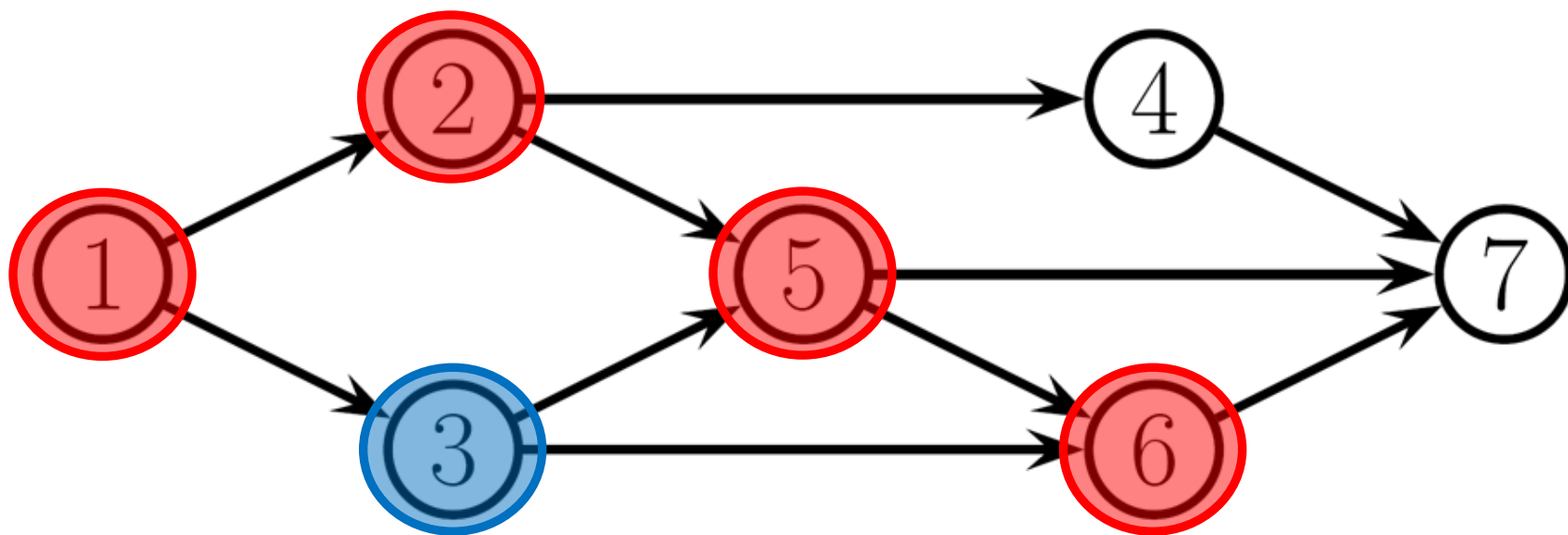
Gibbs sampling



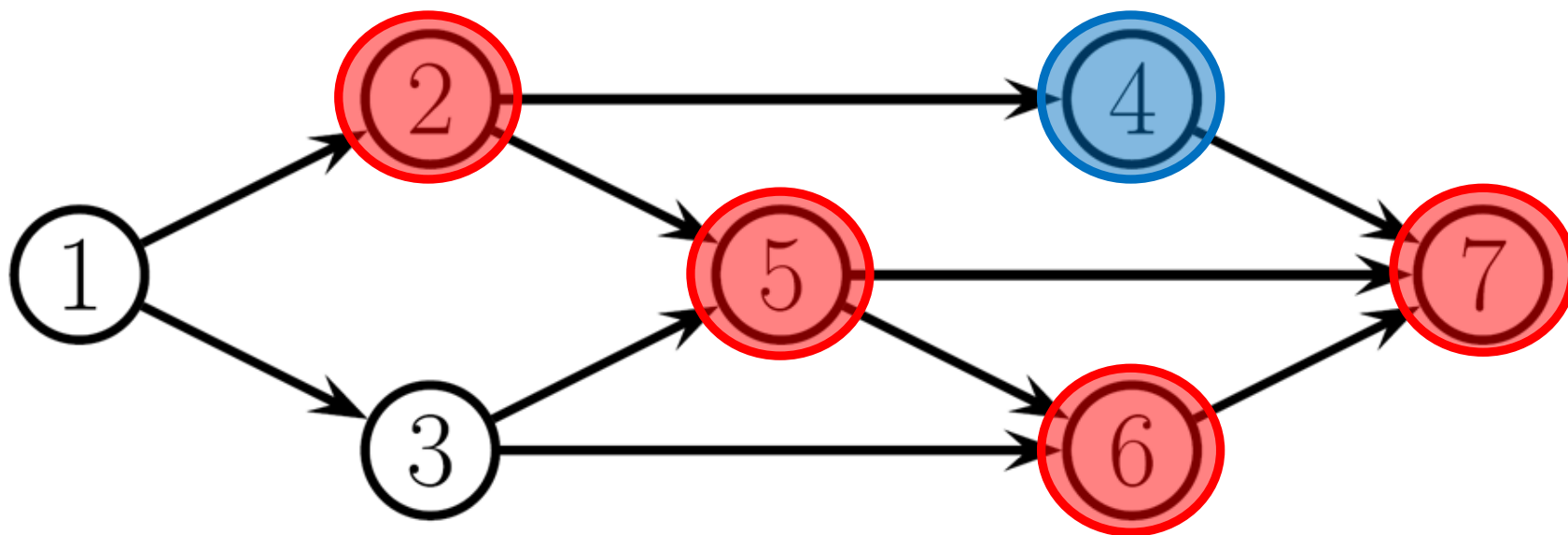
Gibbs sampling



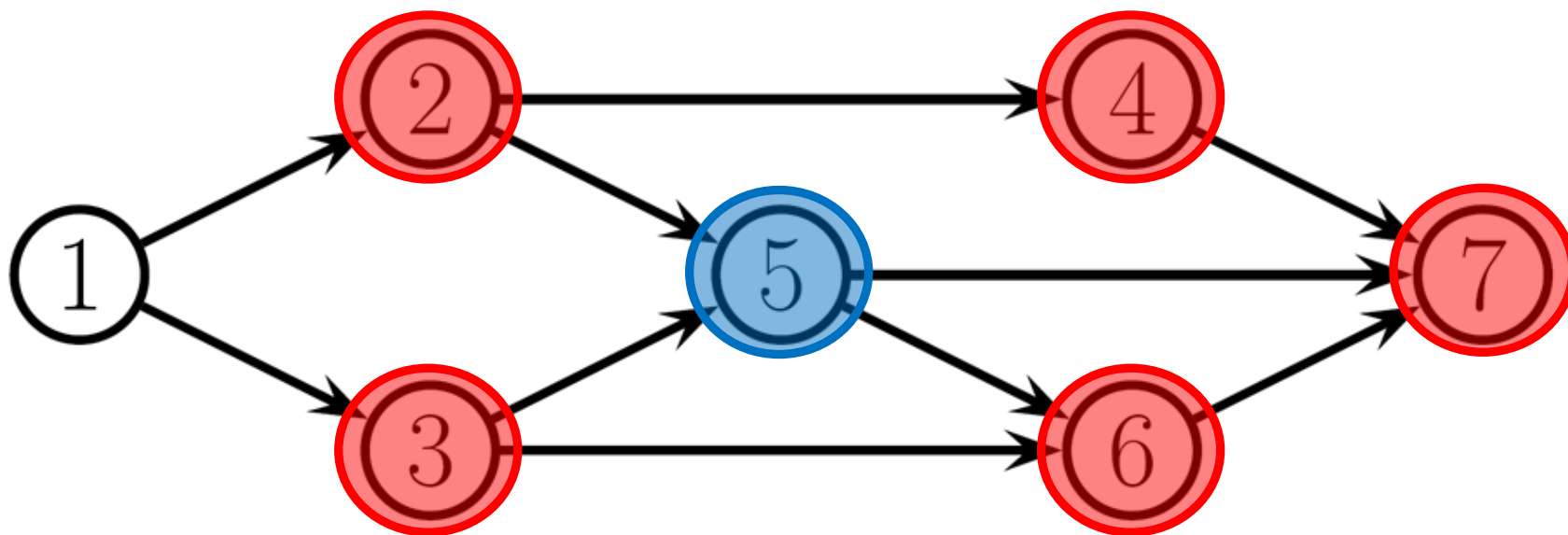
Gibbs sampling



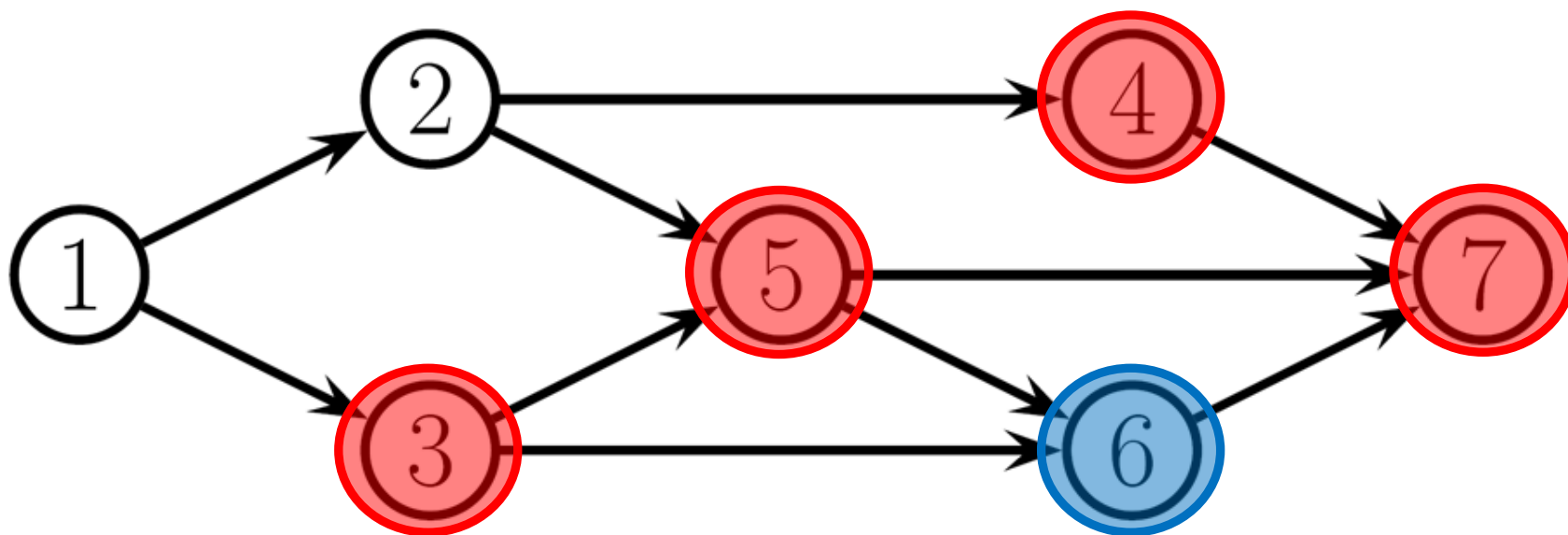
Gibbs sampling



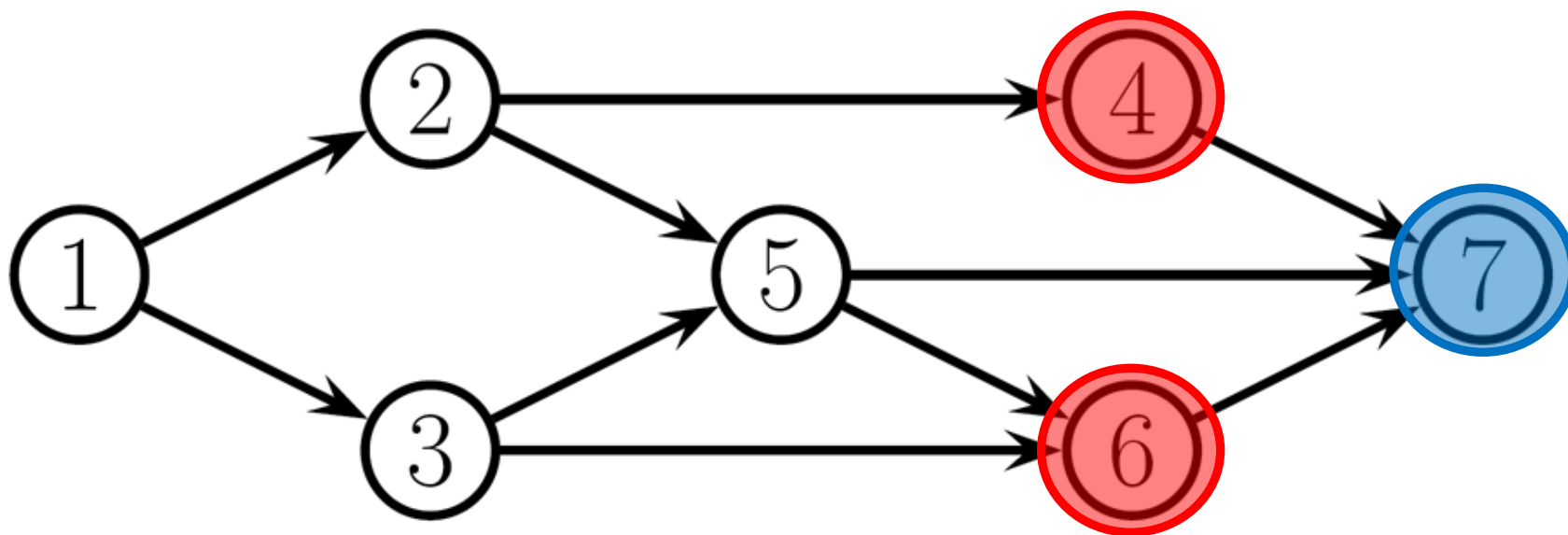
Gibbs sampling



Gibbs sampling



Gibbs sampling



Gibbs sampling for SBM

Initialize group assignments and parameters randomly

Until converged

For each pair of groups k, k'

$$W_{kk'} \sim \text{Beta}(n_{kk'}^{(1)} + \alpha_1, n_{kk'}^{(0)} + \alpha_0)$$

$$\pi \sim \text{Dirichlet}([n_1 + \alpha_1, \dots, n_K + \alpha_K])$$

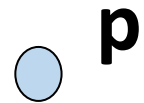
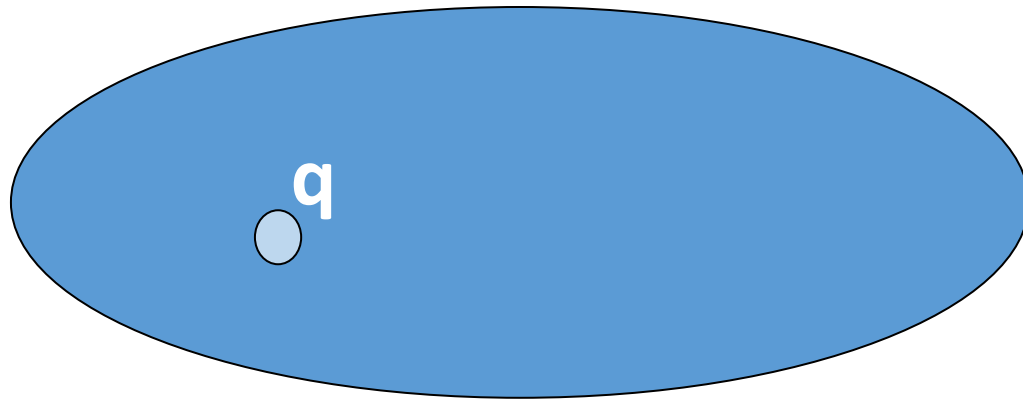
For node i

$$Pr(z_i = k) \propto \pi_k \prod_{k'=1}^K W_{kk'}^{n_{i,k'}^{(1)}} (1 - W_{kk'})^{n_{i,k'}^{(0)}}$$

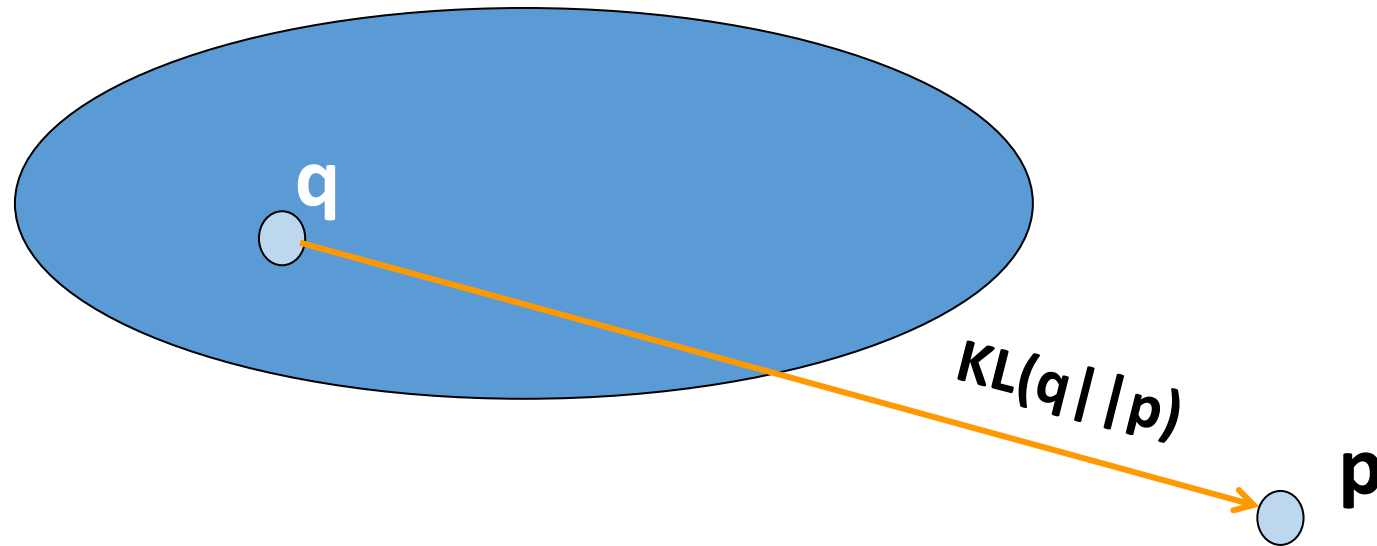
Variational inference

- Key idea:
 - Approximate distribution of interest $p(z)$ with another distribution $q(z)$
 - Make $q(z)$ tractable to work with
 - Solve an optimization problem to make $q(z)$ as similar to $p(z)$ as possible, e.g. in KL-divergence

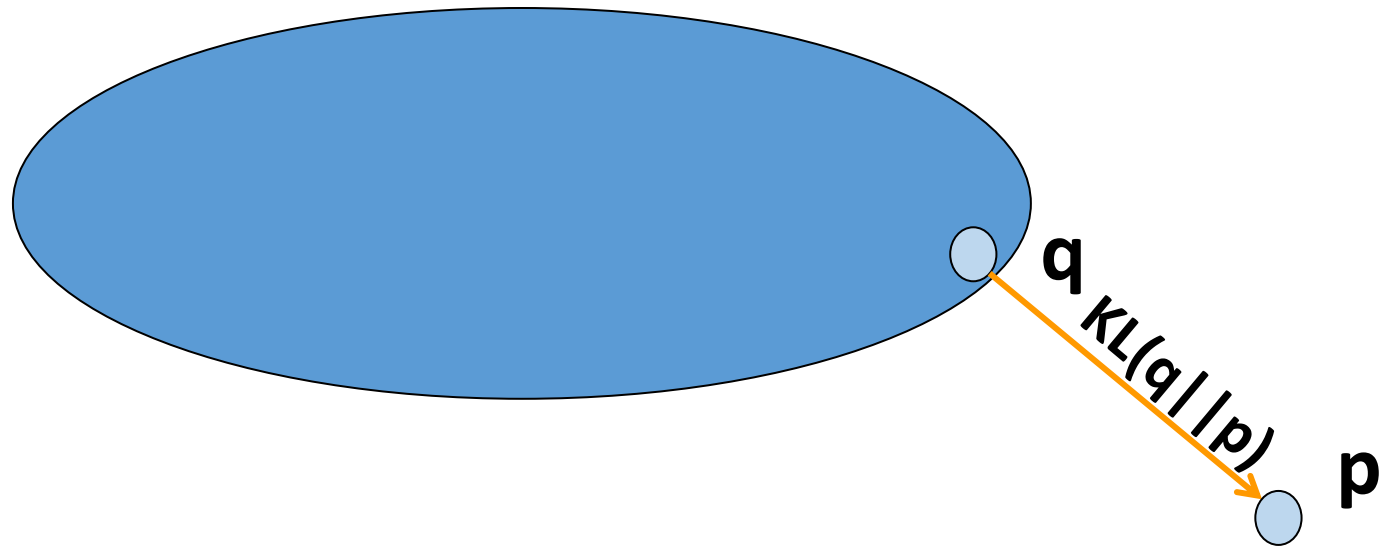
Variational inference



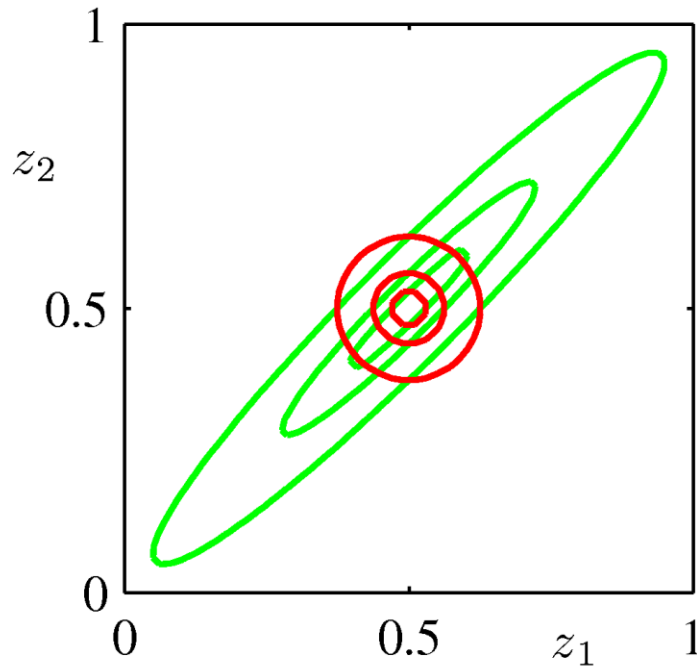
Variational inference



Variational inference

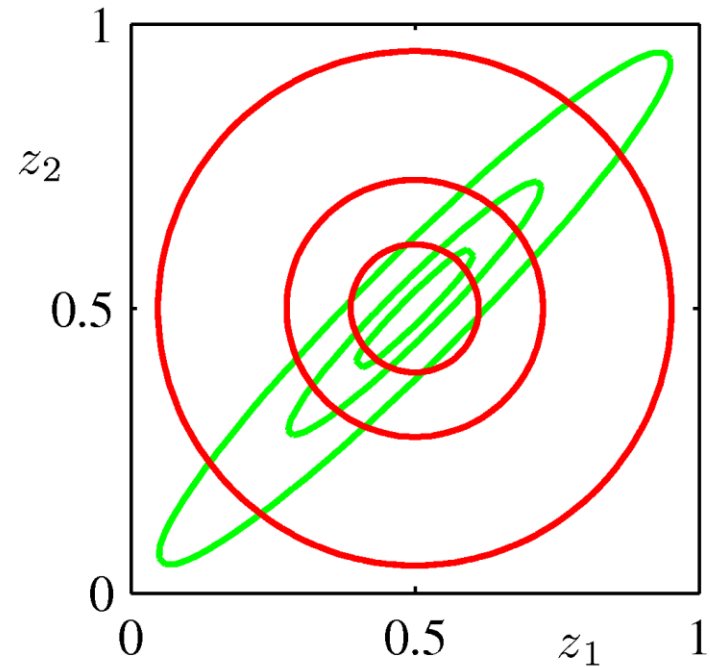


Reverse KL



(a)

Forwards KL



(b)

$$D_{KL}(q(\mathbf{z}) \| p(\mathbf{z})) = \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})}$$

**Blows up if p is small and q isn't.
Under-estimates the support**

$$D_{KL}(p(\mathbf{z}) \| q(\mathbf{z})) = \sum_{\mathbf{z}} p(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})}$$

**Blows up if q is small and p isn't.
Over-estimates the support**

KL-divergence as an objective function for variational inference

$$\begin{aligned} D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) &= E_q \left[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}|\mathbf{x})] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x}) \end{aligned}$$

KL-divergence as an objective function for variational inference

$$\begin{aligned} D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) &= E_q \left[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}|\mathbf{x})] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x}) \end{aligned}$$

- Minimizing the KL is equivalent to maximizing

$$\mathcal{L}(q) = E_q[\log p(\mathbf{z}, \mathbf{x})] - E_q[\log q(\mathbf{z})]$$

KL-divergence as an objective function for variational inference

$$\begin{aligned} D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) &= E_q \left[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}|\mathbf{x})] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x}) \end{aligned}$$

- Minimizing the KL is equivalent to maximizing

$$\begin{aligned} \mathcal{L}(q) &= E_q[\log p(\mathbf{z}, \mathbf{x})] - E_q[\log q(\mathbf{z})] \\ &= E_q[\log p(\mathbf{z}, \mathbf{x})] + H[q] \end{aligned}$$

KL-divergence as an objective function for variational inference

$$\begin{aligned} D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) &= E_q \left[\log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}|\mathbf{x})] \\ &= E_q[\log q(\mathbf{z})] - E_q[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x}) \end{aligned}$$

- Minimizing the KL is equivalent to maximizing

$$\begin{aligned} \mathcal{L}(q) &= E_q[\log p(\mathbf{z}, \mathbf{x})] - E_q[\log q(\mathbf{z})] \\ &= E_q[\log p(\mathbf{z}, \mathbf{x})] + H[q] \end{aligned}$$

Fit the data well



Be flat



Mean field variational inference

- We still need to compute expectations over $\mathbf{z}$
- However, we have gained the option to restrict $q(\mathbf{z})$ to make these expectations tractable.
- The mean field approach uses a fully factorized $q(\mathbf{z})$

$$q(\mathbf{z}) = \prod_i q_i(z_i)$$

The entropy term decomposes nicely:

$$-E_q[\log q(\mathbf{z})] = -E_q[\log \prod_i q_i(z_i)] = \sum_i E_{q_i}[-\log q_i(z_i)] = \sum_i H(q_i)$$

Mean field algorithm

- Until converged
 - For each factor i
 - Select variational parameters γ_i such that

$$q_i(z_i | \gamma_i) \propto \exp(E_{q_{\neg i}}[\log p(\mathbf{z}, \mathbf{x})])$$

- Each update monotonically improves the ELBO so the algorithm must converge

Deriving mean field updates for your model

- Write down the mean field equation explicitly,

$$\log q_i(z_i) := E_{q_{\neg i}} [\log p(\mathbf{z}, \mathbf{x})] + \text{const}$$

- Simplify and apply the expectation.
- Manipulate it until you can recognize it as a log-pdf of a known distribution (hopefully).
- Reinststate the normalizing constant.

Mean field vs Gibbs sampling

- Both mean field and Gibbs sampling iteratively update one variable given the rest
- Mean field stores an entire distribution for each variable, while Gibbs sampling draws from one.

Pros and cons vs Gibbs sampling

- **Pros:**

- Deterministic algorithm, typically converges faster
- Stores an analytic representation of the distribution, not just samples
- Non-approximate parallel algorithms
- Stochastic algorithms can scale to very large data sets
- No issues with checking convergence

- **Cons:**

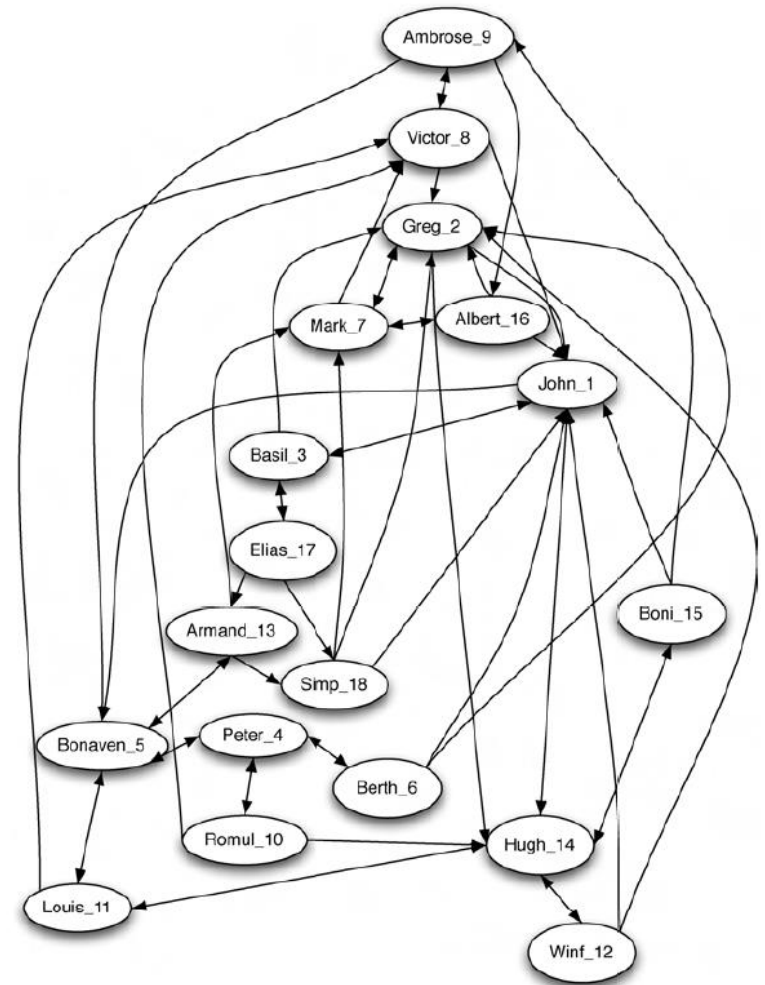
- Will never converge to the true distribution, unlike Gibbs sampling
- Dense representation can mean more communication for parallel algorithms
- Harder to derive update equations

Variational inference algorithm for MMSB (Variational EM)

- Compute maximum likelihood estimates for interaction parameters $W_{kk'}$
- Assume fully factorized variational distribution for mixed membership vectors, cluster assignments
- Until converged
 - For each node
 - Compute variational discrete distribution over it's latent $z_{p \rightarrow q}$ and $z_{q \rightarrow p}$ assignments
 - Compute variational Dirichlet distribution over its mixed membership distribution
 - Maximum likelihood update for $\mathbf{W}$

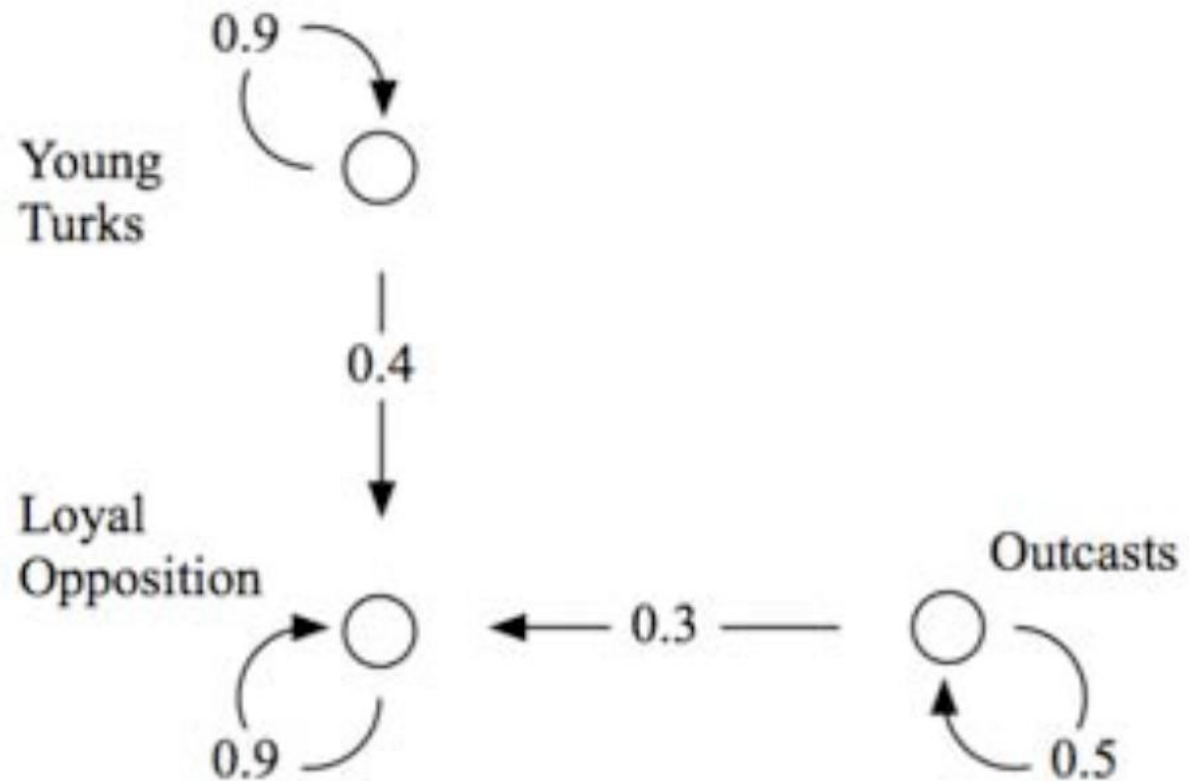
Application of MMSB to Sampson's Monastery

- Sampson (1968) studied friendship relationships between novice monks
- Identified several factions
 - Blockmodel appropriate?
- Conflicts occurred
 - Two monks expelled
 - Others left



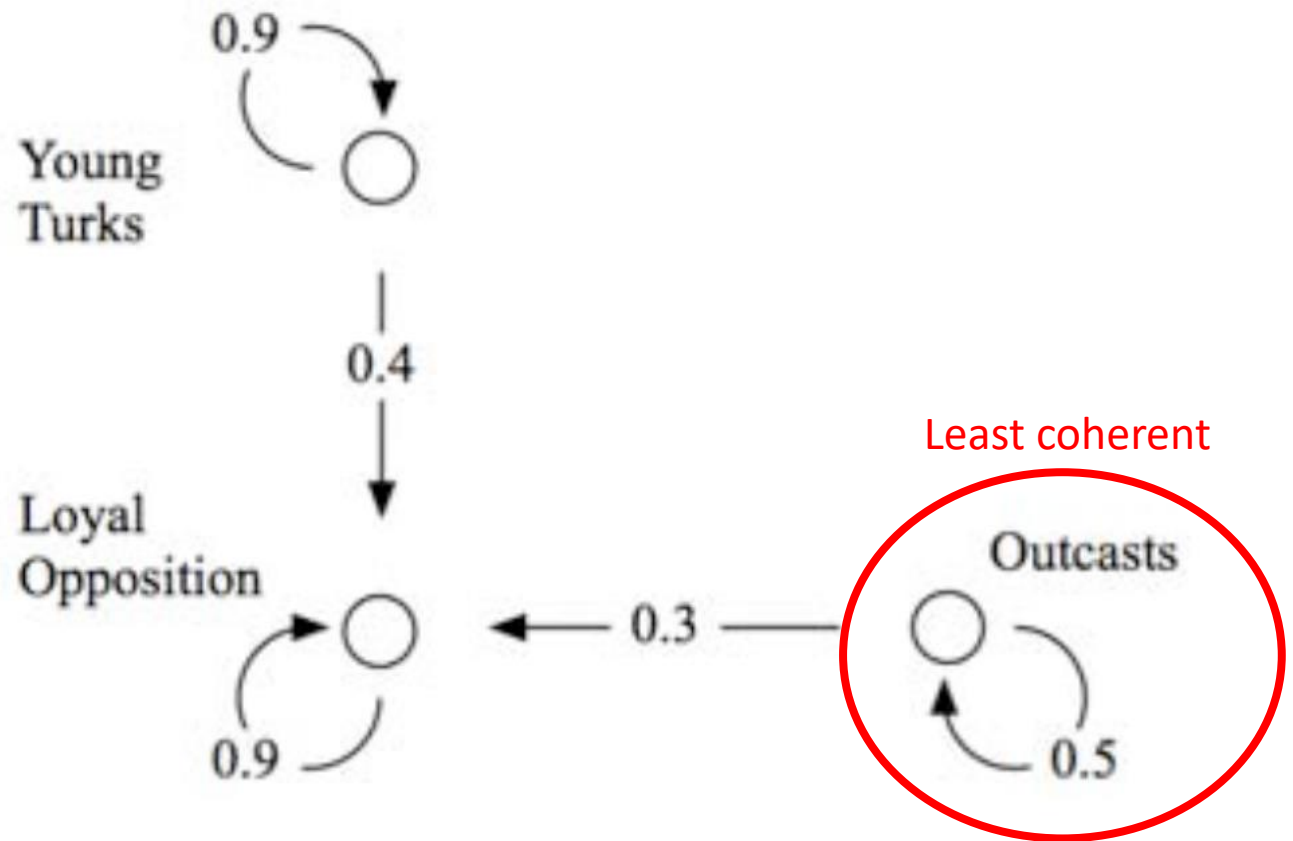
Application of MMSB to Sampson's Monastery

**Estimated
blockmodel**



Application of MMSB to Sampson's Monastery

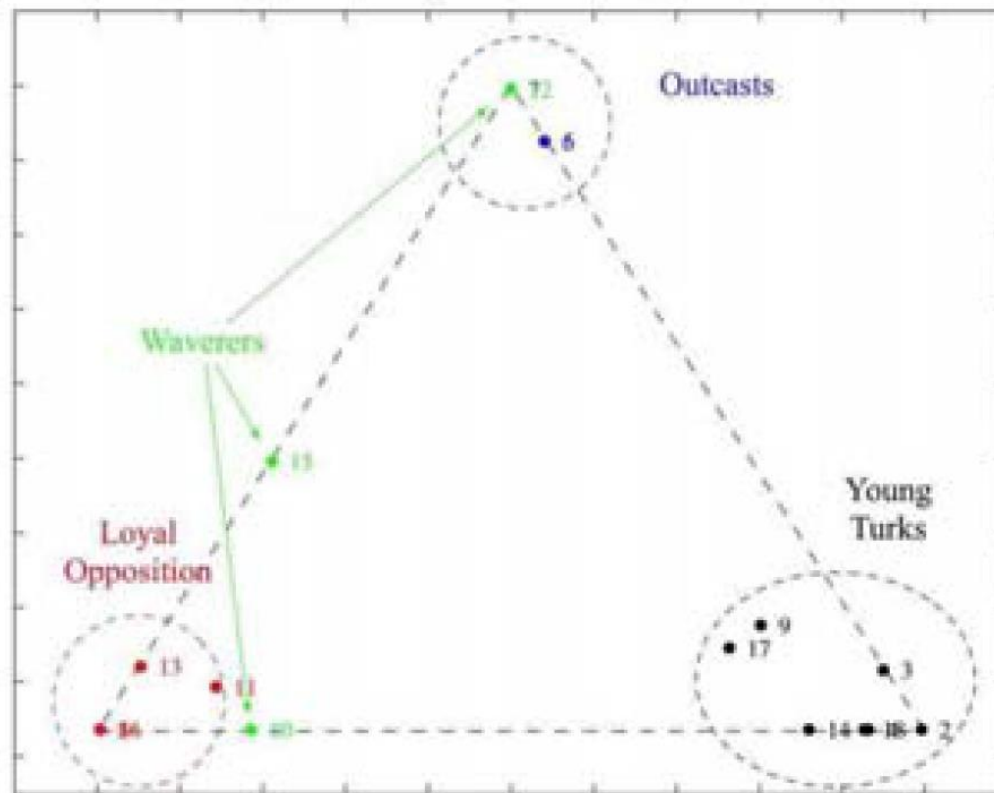
**Estimated
blockmodel**



Application of MMSB to Sampson's Monastery

Estimated Mixed
membership
vectors

(posterior mean)

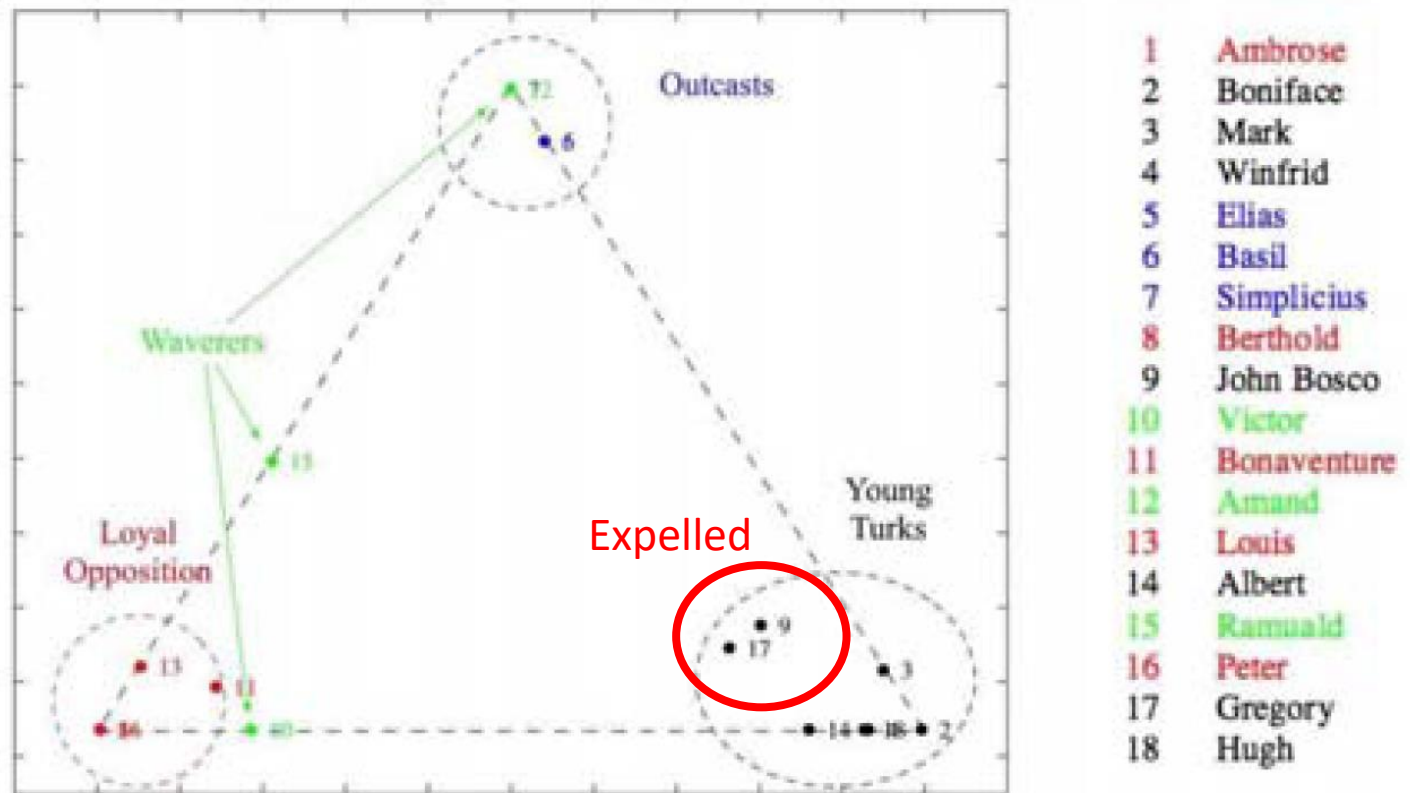


- | | |
|----|-------------|
| 1 | Ambrose |
| 2 | Boniface |
| 3 | Mark |
| 4 | Winfred |
| 5 | Elias |
| 6 | Basil |
| 7 | Simplicius |
| 8 | Berthold |
| 9 | John Bosco |
| 10 | Victor |
| 11 | Bonaventure |
| 12 | Amand |
| 13 | Louis |
| 14 | Albert |
| 15 | Ramuald |
| 16 | Peter |
| 17 | Gregory |
| 18 | Hugh |

Application of MMSB to Sampson's Monastery

Estimated Mixed
membership
vectors

(posterior mean)

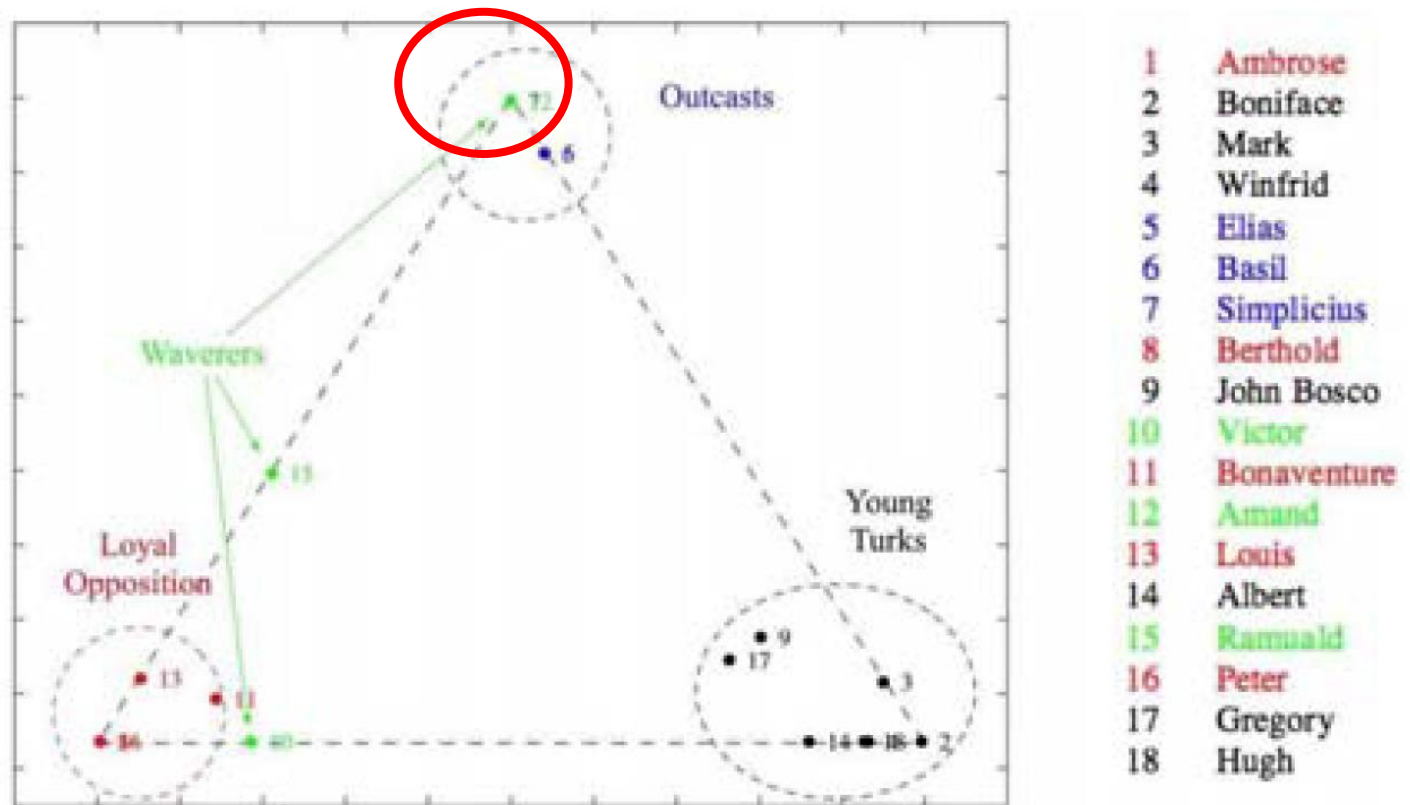


Application of MMSB to Sampson's Monastery

Wavering not captured

Estimated Mixed
membership
vectors

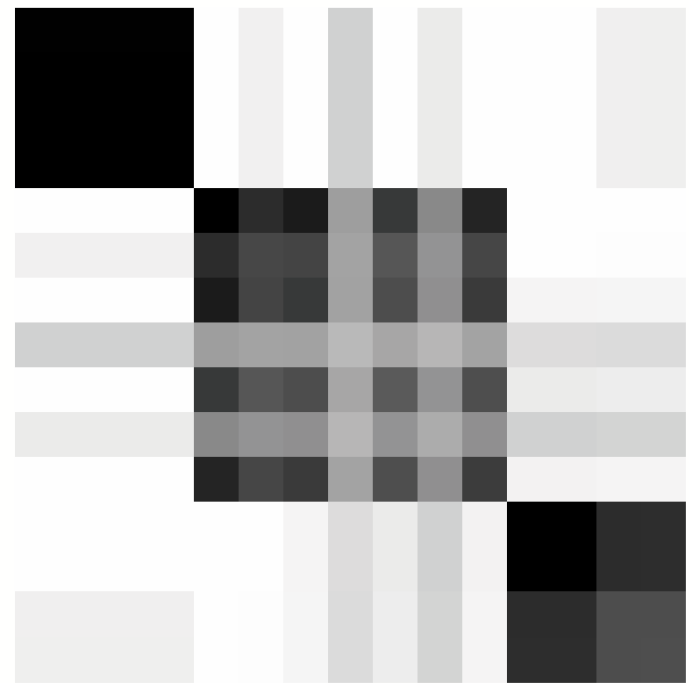
(posterior mean)



Application of MMSB to Sampson's Monastery



Original network
(whom do you like?)



Summary of network (use π 's)

Application of MMSB to Sampson's Monastery



Original network
(whom do you like?)



Denoise network (use z 's)

Scaling up Bayesian inference to large networks

- Two key strategies: **parallel/distributed**, and **stochastic algorithms**
- **Parallel/distributed algorithms**
 - Compute VB or MCMC updates in parallel
 - Communication overhead may be lower for MCMC
 - Not well understood for MCMC, but works in practice
- **Stochastic algorithms**
 - Stochastic variational inference
 - estimate updates based on subsamples. MMSB: *Gopalan et al. (2012)*
 - A related subsampling trick for MCMC in latent space models (*Raftery et al., 2012*)
 - Other general stochastic MCMC algorithms:
 - Stochastic gradient Langevin dynamics (Welling and Teh, 2011), Austerity MCMC (Korattika et al., 2014)

Evaluation of unsupervised models

- **Quantitative** evaluation
 - Measurable, quantifiable performance metrics
- **Qualitative** evaluation
 - Exploratory data analysis (EDA) using the model
 - Human evaluation, user studies,...

Evaluation of unsupervised models

- **Intrinsic** evaluation
 - Measure inherently good properties of the model
 - Fit to the data (e.g. link prediction), interpretability,...
- **Extrinsic** evaluation
 - Study usefulness of model for external tasks
 - Classification, retrieval, part of speech tagging,...

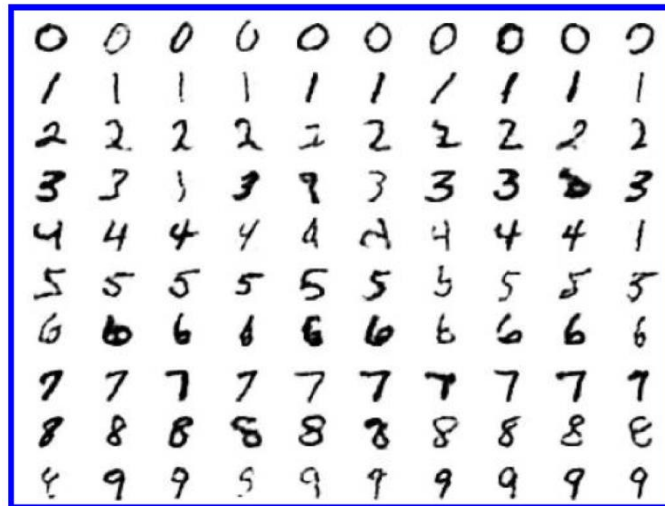
Extrinsic evaluation:

What will you use your model for?

- If you have a **downstream task** in mind, you should probably evaluate based on it!
- Even if you don't, you could contrive one for evaluation purposes
- E.g. use latent representations for:
 - Classification, regression, retrieval, ranking...

Posterior predictive checks

- Sampling data from the posterior predictive distribution allows us to “look into the mind of the model” – G. Hinton



“This use of the word *mind* is not intended to be metaphorical. We believe that a mental state is the state of a hypothetical, external world in which a high-level internal representation would constitute veridical perception. That hypothetical world is what the figure shows.” **Geoff Hinton et al. (2006), A Fast Learning Algorithm for Deep Belief Nets.**

Posterior predictive checks

- Does data drawn from the model differ from the observed data, in ways that we care about?
- PPC:
 - Define a discrepancy function (a.k.a. test statistic) $T(\mathbf{X})$.
 - Like a test statistic for a p-value. How extreme is my data set?
 - Simulate new data $\mathbf{X}^{(rep)}$ from the posterior predictive
 - Use MCMC to sample parameters from posterior, then simulate data
 - Compute $T(\mathbf{X}^{(rep)})$ and $T(\mathbf{X})$, compare. Repeat, to estimate:

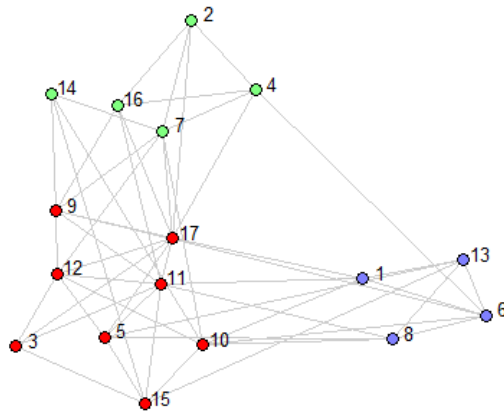
$$PPC = P(T(\mathbf{X}^{(rep)}) > T(\mathbf{X}) | \mathbf{X})$$

Outline

- Mathematical representations of social networks and generative models
 - Introduction to generative approach
 - Connections to sociological principles
- Fitting generative social network models to data
 - Example application scenarios
 - Model selection and evaluation
- Recent developments in generative social network models
 - Dynamic social network models

Dynamic social network

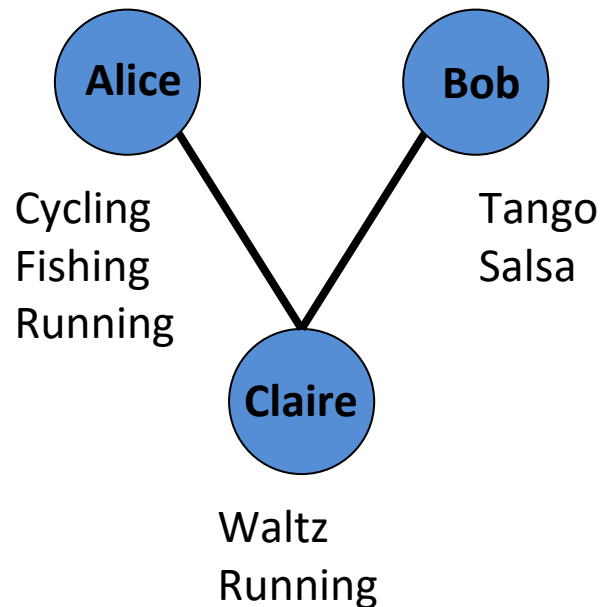
- Relations between people may change over time
- Need to generalize social network models to account for dynamics



Dynamic social network
(Nordlie, 1958; Newcomb, 1961)

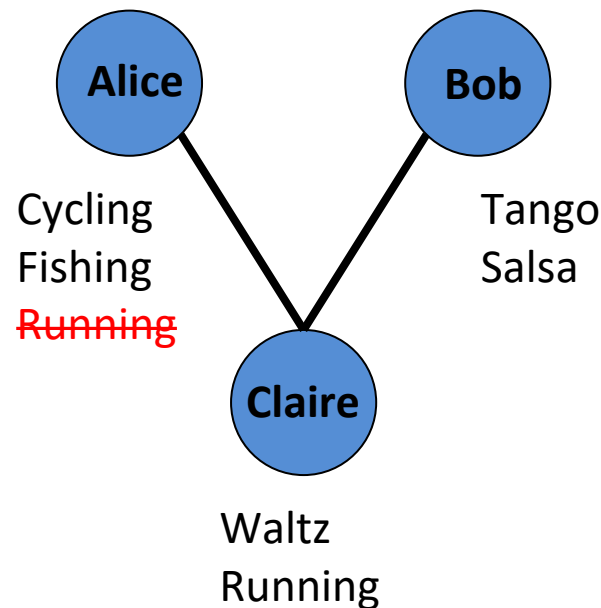
Dynamic Relational Infinite Feature Model (DRIFT)

- Models networks as they over time, by way of changing latent features



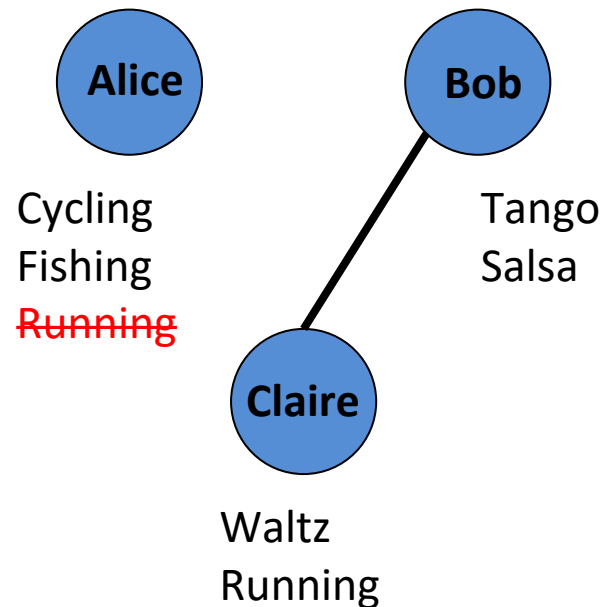
Dynamic Relational Infinite Feature Model (DRIFT)

- Models networks as they over time, by way of changing latent features



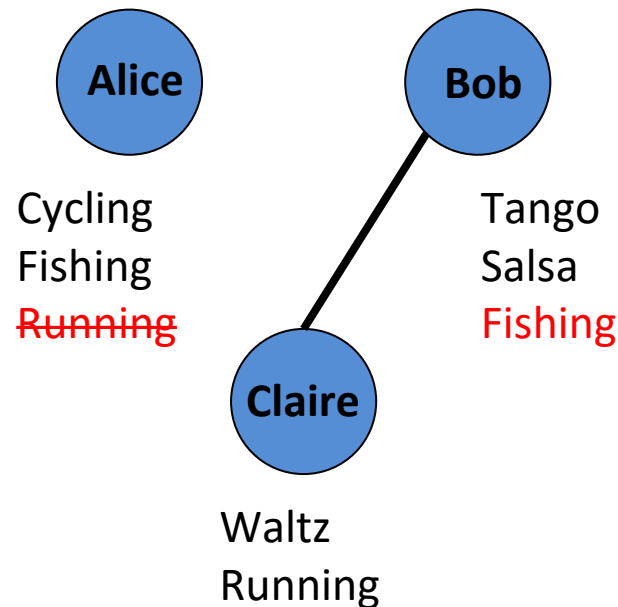
Dynamic Relational Infinite Feature Model (DRIFT)

- Models networks as they over time, by way of changing latent features



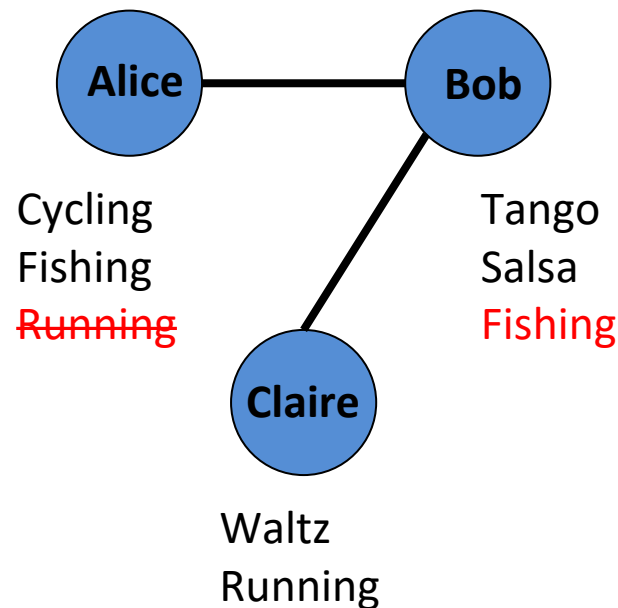
Dynamic Relational Infinite Feature Model (DRIFT)

- Models networks as they over time, by way of changing latent features



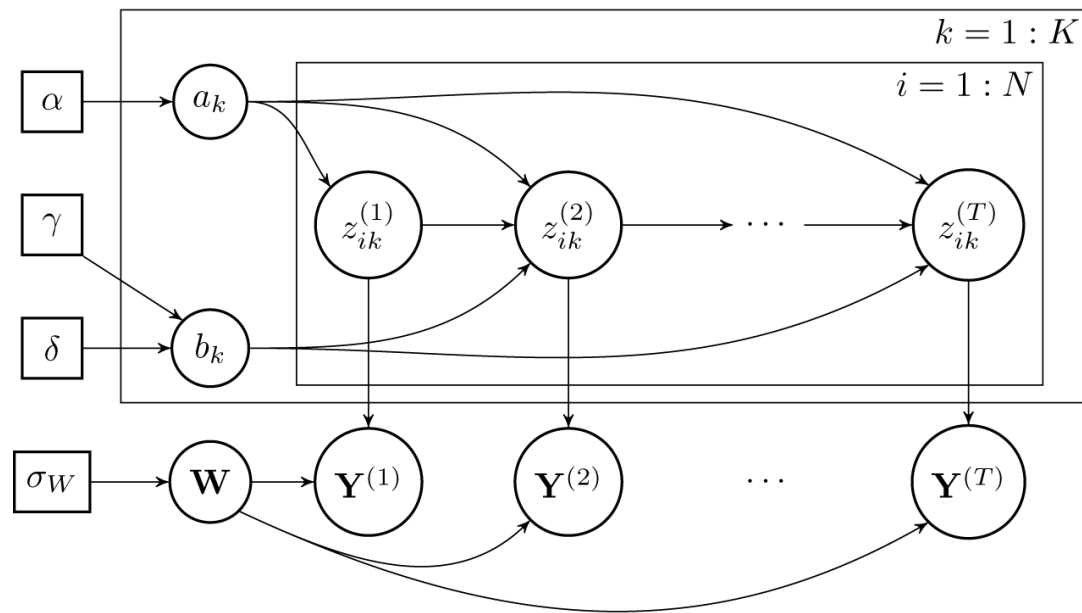
Dynamic Relational Infinite Feature Model (DRIFT)

- Models networks as they over time, by way of changing latent features



Dynamic Relational Infinite Feature Model (DRIFT)

- Models networks as they over time, by way of changing latent features

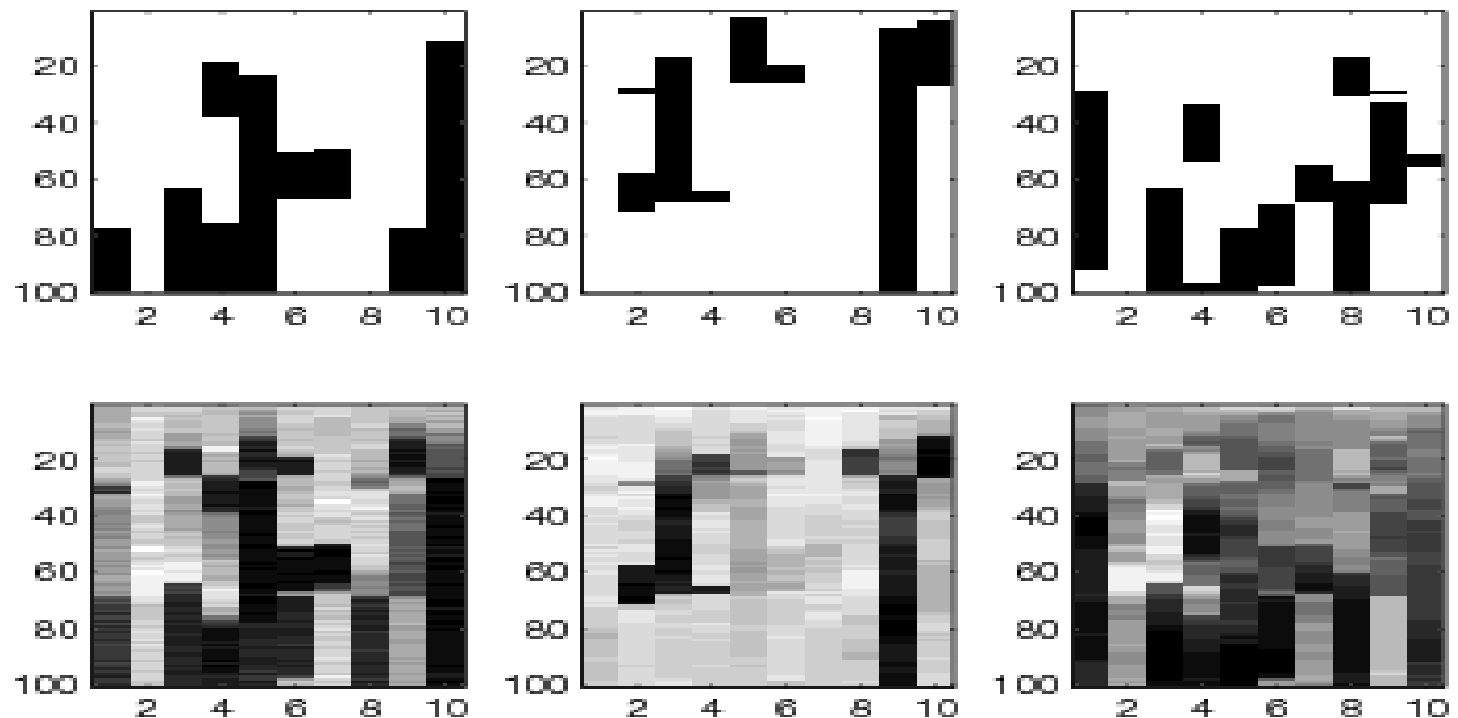


- HMM dynamics for each actor/feature (factorial HMM)

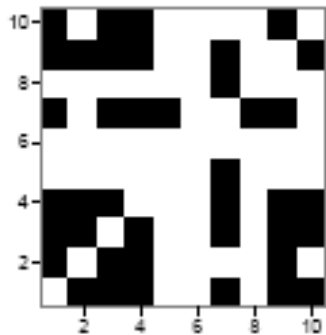
Bayesian Inference for DRIFT

- Markov chain Monte Carlo inference
 - Blocked Gibbs sampler
 - Forward filtering, backward sampling to jointly sample each actor's feature chains
 - “Slice sampling” trick with the stick-breaking construction of the IBP to adaptively truncate the number of features but still perform exact inference
 - Metropolis-Hastings updates for W 's

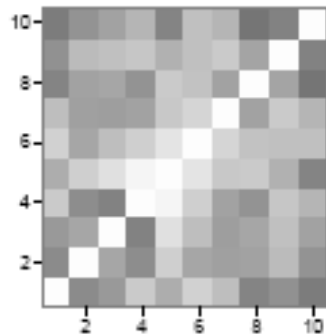
Synthetic Data: Inference on Z 's



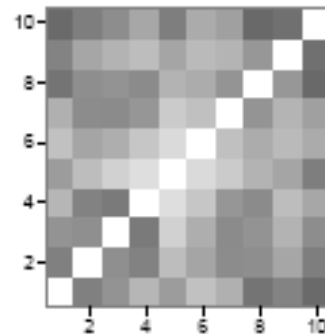
Synthetic Data: Predicting the Future



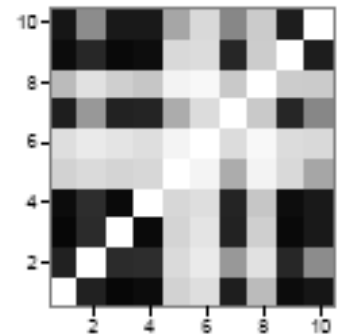
(a) True Y



(b) Baseline



(c) RIFT (all)



(d) DRIFT

Enron Email Data: Predicting the Future

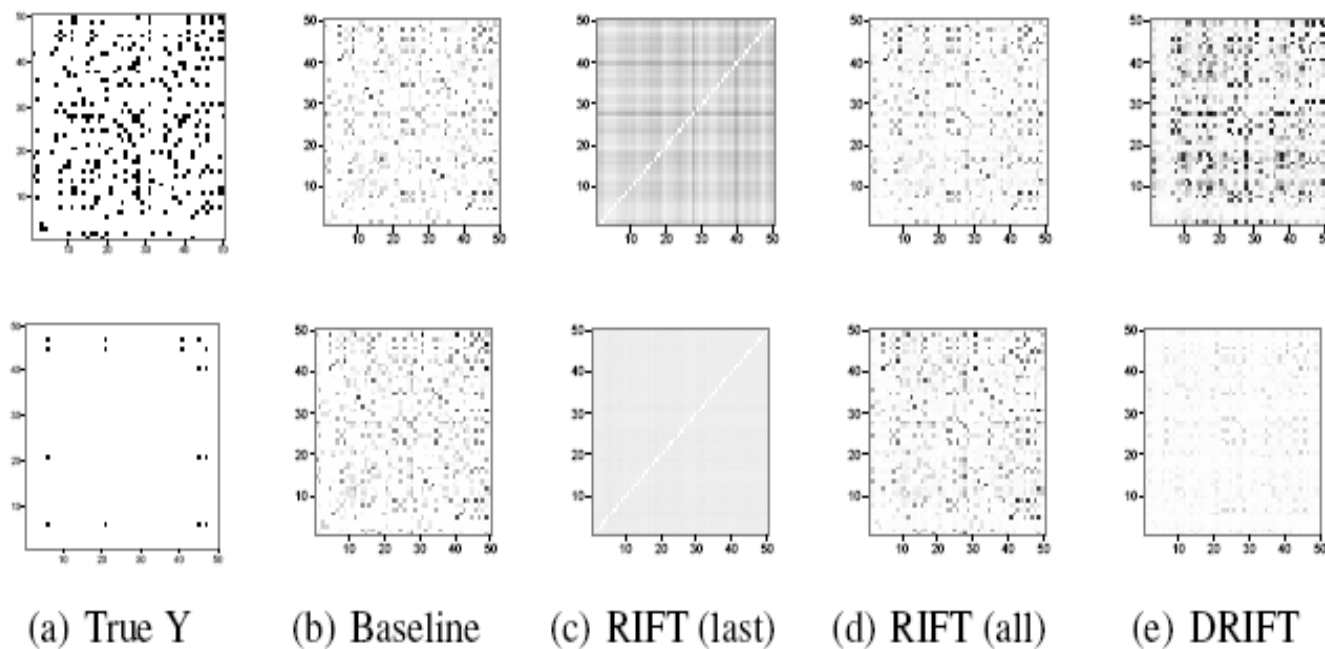
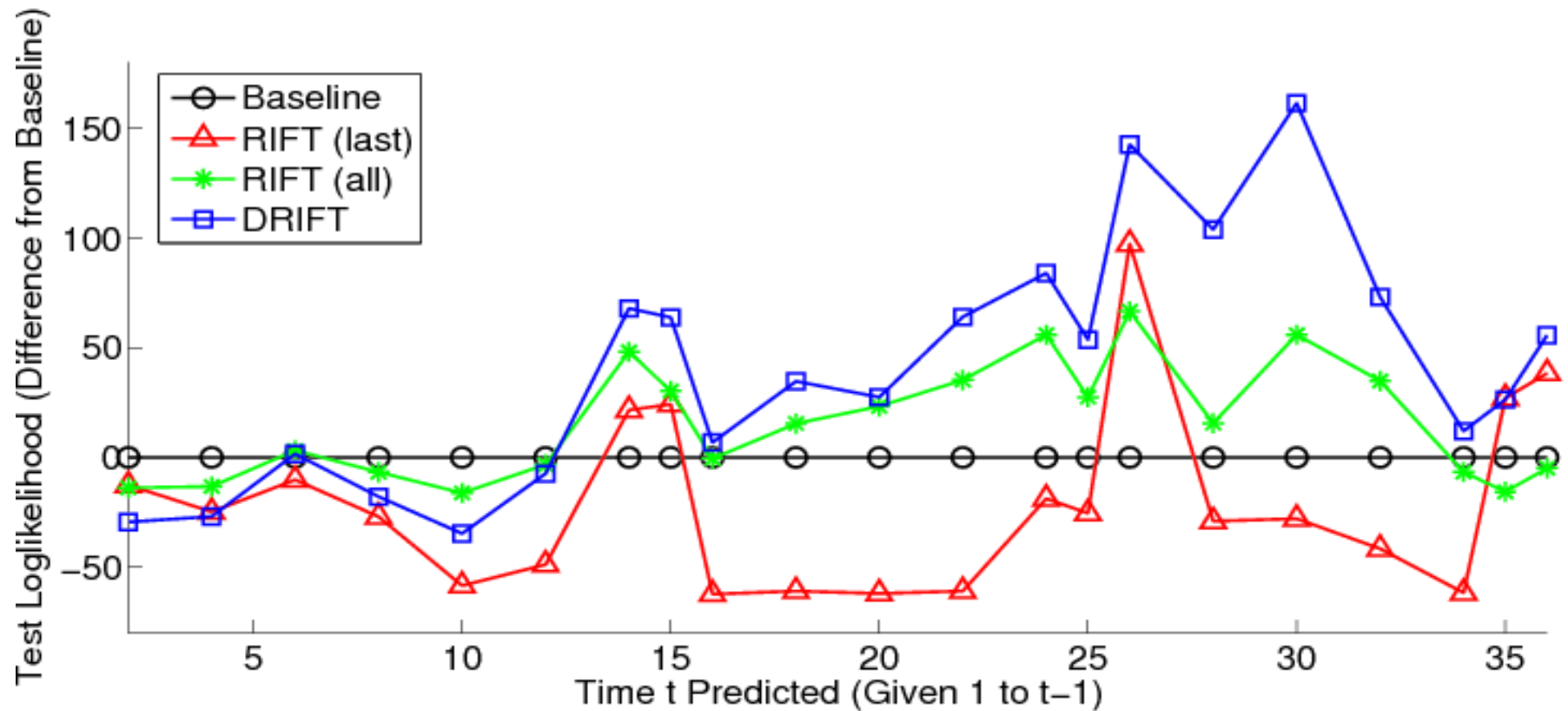
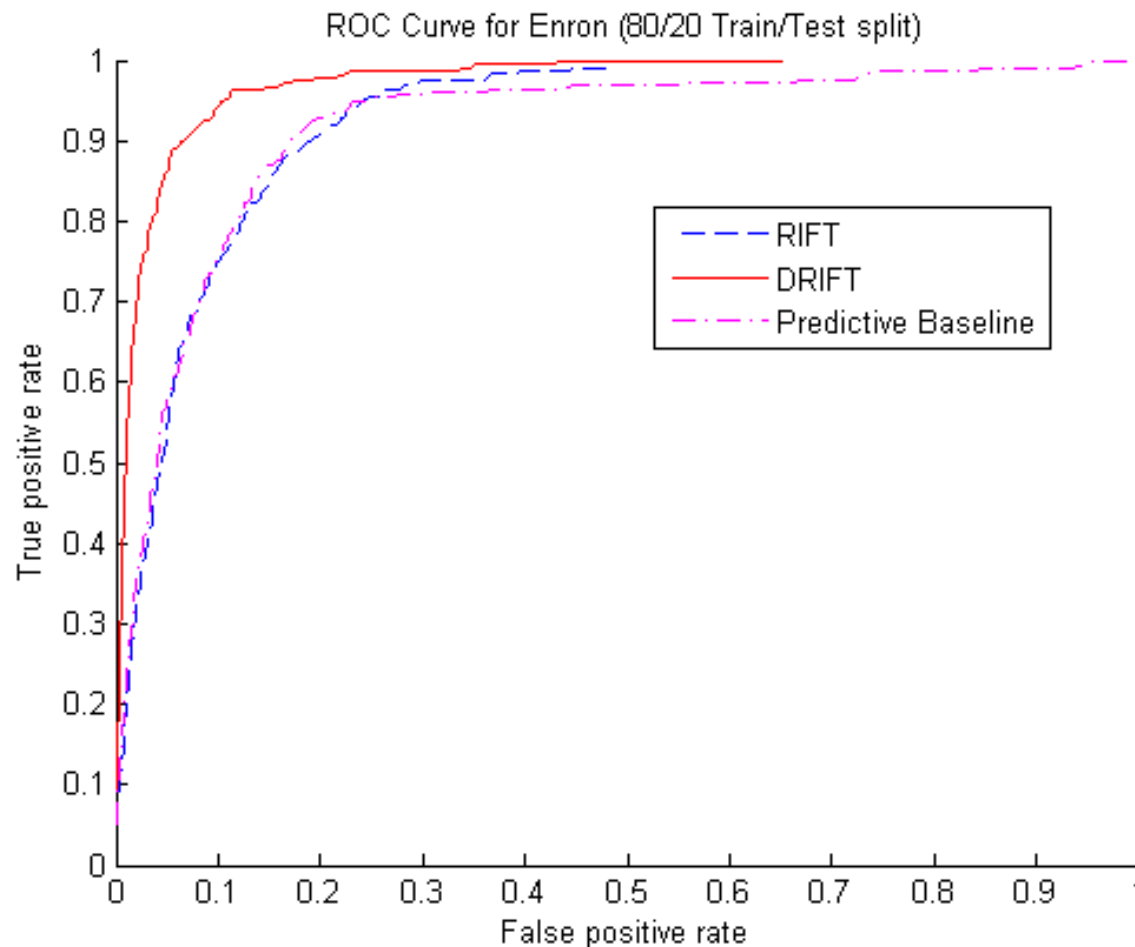


Figure 5: Held out Y at time $t=30$ (top row) and $t=36$ (bottom row) for Enron, and posterior predictive distributions for each of the methods.

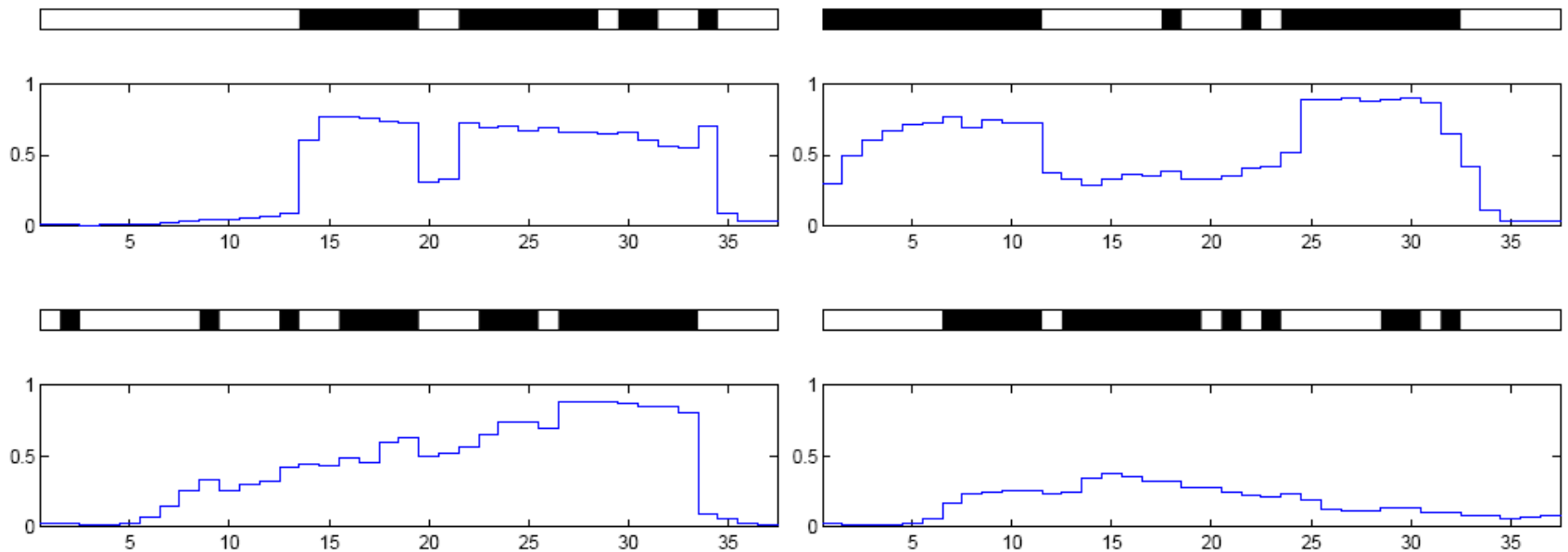
Enron Email Data: Predicting the Future



Enron Email Data: Missing Data Imputation



Enron Email Data: Edge Probability Over Time

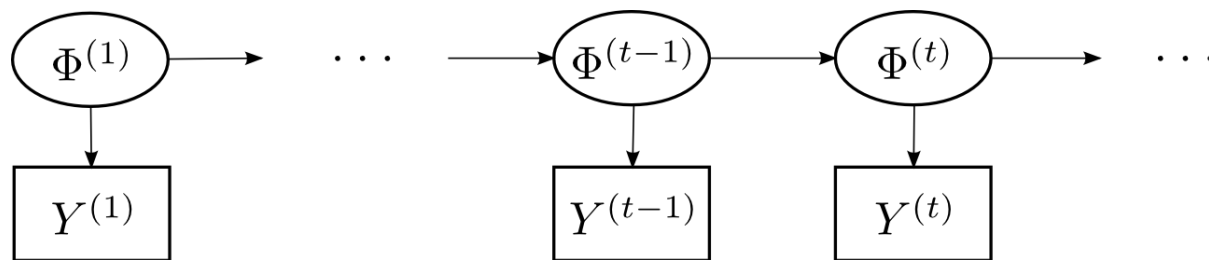


Quantitative Results

Synthetic Dataset	Naive	Baseline	LFRM (last/current)	LFRM (all)	DRIFT
Forecast LL	-31.6	-32.6	-28.4	-31.6	-11.6
Missing Data LL	-575	-490	-533	-478	-219
Forecast AUC	N/A	0.608	0.779	0.596	0.939
Missing Data AUC	N/A	0.689	0.675	0.691	0.925
Enron Dataset	Naive	Baseline	LFRM (last/current)	LFRM (all)	DRIFT
Forecast LL	-141	-108	-119	-98.3	-83.5
Missing Data LL	-1610	-1020	-1410	-981	-639
Forecast AUC	N/A	0.874	0.777	0.891	0.910
Missing Data AUC	N/A	0.921	0.803	0.933	0.979

Hidden Markov dynamic network models

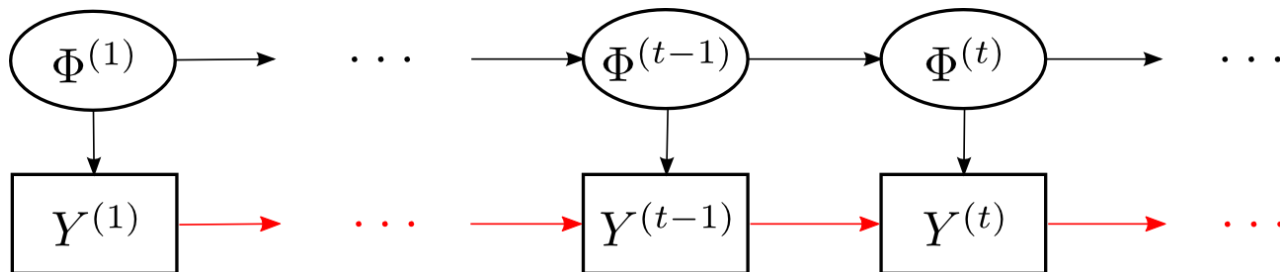
- Most work on dynamic network modeling assumes **hidden Markov** structure
 - Latent variables and/or parameters follow Markov dynamics
 - Graph snapshot at each time generated using **static network model**, e.g. stochastic block model or latent feature model as in DRIFT



- Has been used to extend SBMs to dynamic models (Yang et al., 2011; Xu and Hero, 2014)

Beyond hidden Markov networks

- Hidden Markov structure is tractable but not very realistic assumption in social interaction networks
 - Interaction between two people **does not** influence future interactions
- Proposed model: Allow current graph to depend on **current parameters** and **previous graph**



- Proposed inference procedure does **not** require MCMC
 - Scales to ~ 1000 nodes

Stochastic block transition model

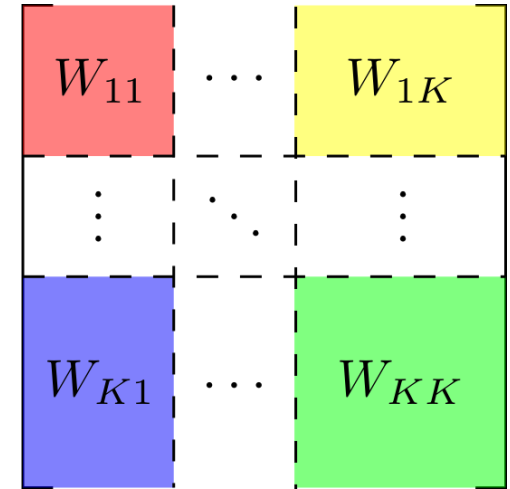
- Main idea: parameterize each block (k, k') with **two** probabilities

- Probability of forming **new** edge

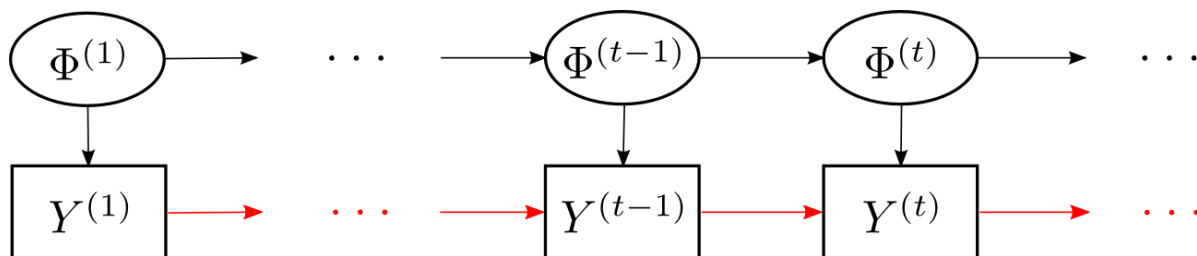
$$\pi_{kk'}^{t|0} = \Pr(Y_{ij}^{(t)} = 1 | Y_{ij}^{(t-1)} = 0)$$

- Probability of **existing** edge re-occurring

$$\pi_{kk'}^{t|1} = \Pr(Y_{ij}^{(t)} = 1 | Y_{ij}^{(t-1)} = 1)$$

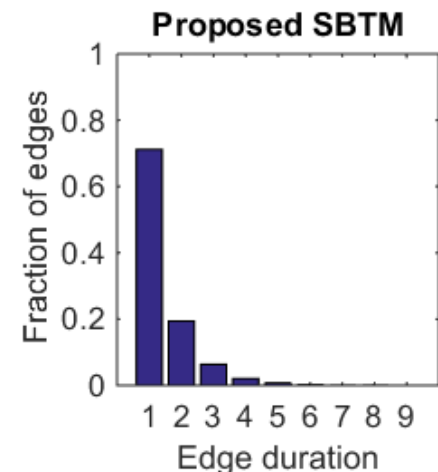
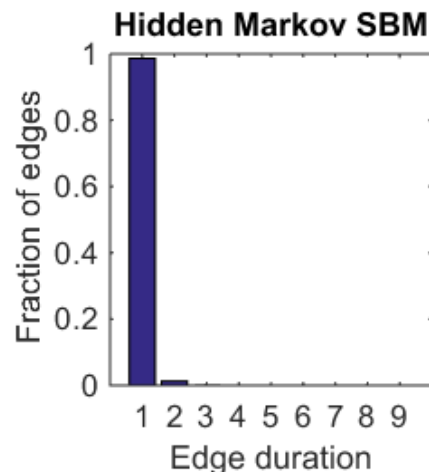
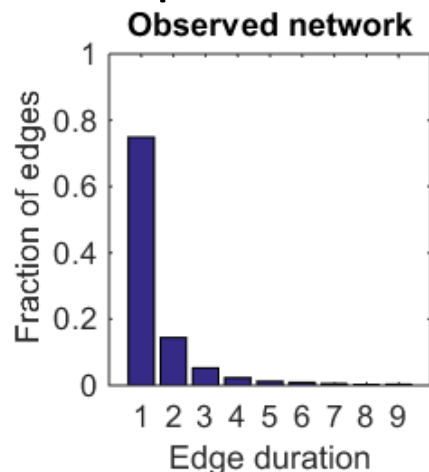


- Generate graph at initial time step using SBM
- Place Markov model on $\Pi^{t|0}, \Pi^{t|1}$



Application to Facebook wall posts

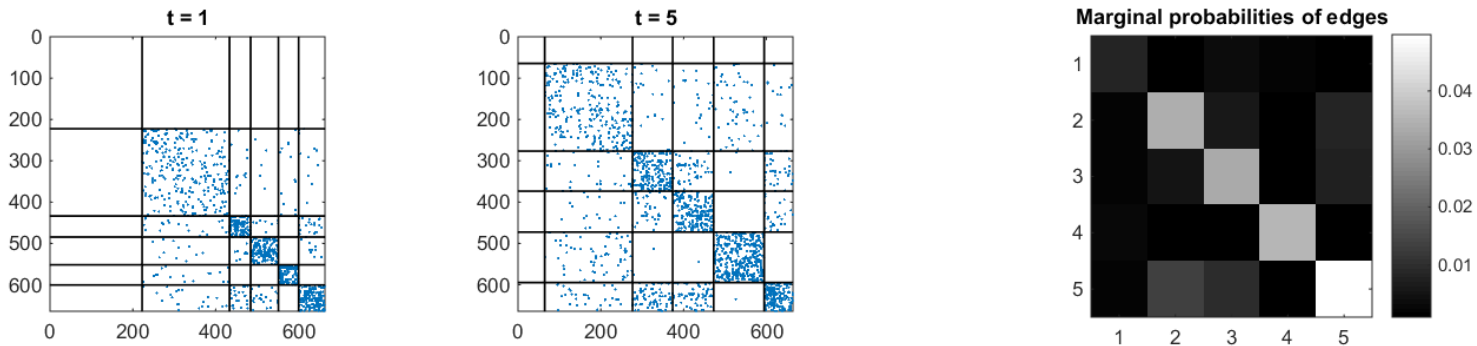
- Fit dynamic SBMs to network of Facebook wall posts
 - ~ 700 nodes, 9 time steps, 5 classes
- How accurately do hidden Markov SBM and SBTM replicate **edge durations** in observed network?
 - Simulate networks from both models using estimated parameters



- Hidden Markov SBM **cannot** replicate long-lasting edges in sparse blocks

Behaviors of different classes

- SBTM retains **interpretability** of SBM at each time step

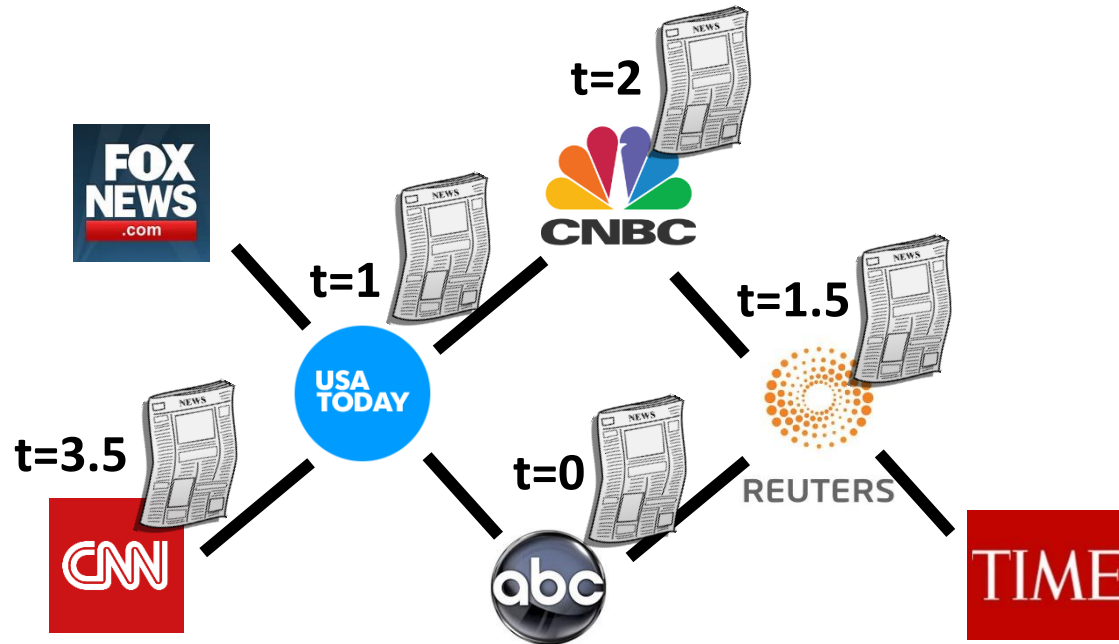


- Q: Do different classes behave differently in how they form edges?



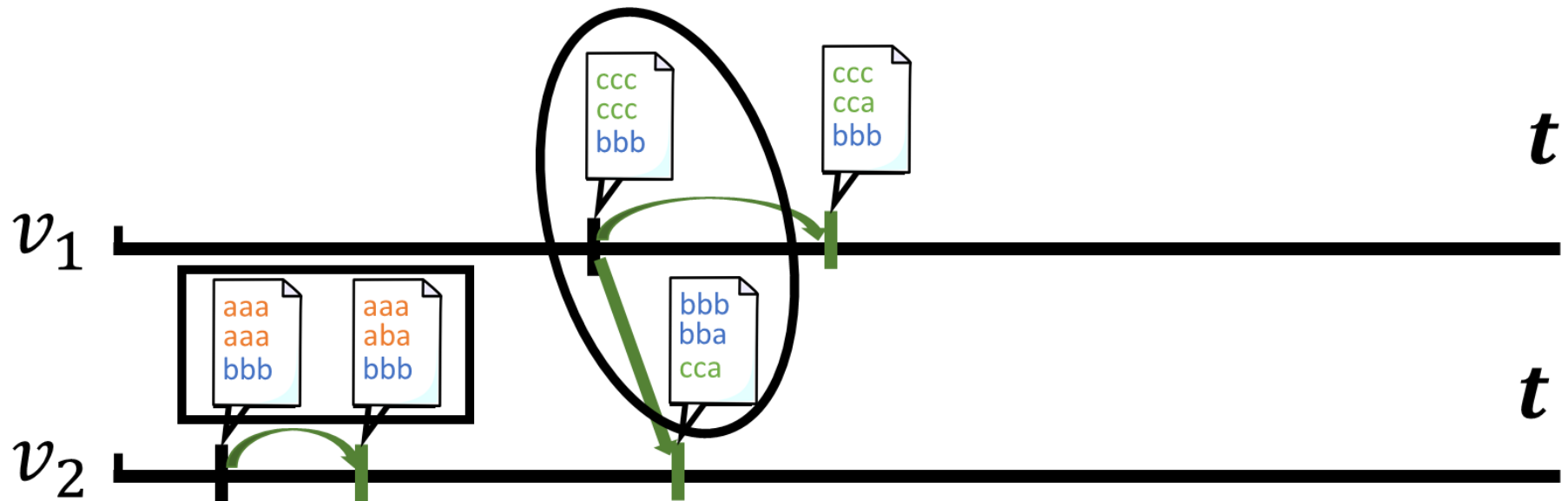
- A: Only for probability of **existing** edges re-occurring
- New insight** revealed by having separate probabilities in SBTM

Information diffusion in text-based cascades



- Temporal information
- Content information
- Network is latent

HawkesTopic model for text-based cascades



Mutual exciting nature: A posting event can trigger future events

Content cascades: The content of a document should be similar to the document that triggers its publication

Modeling posting times

Mutually exciting nature captured via **Multivariate Hawkes Process** (MHP) [Liniger 09].

For MHP, **intensity process** $\lambda_v(t)$ takes the form:

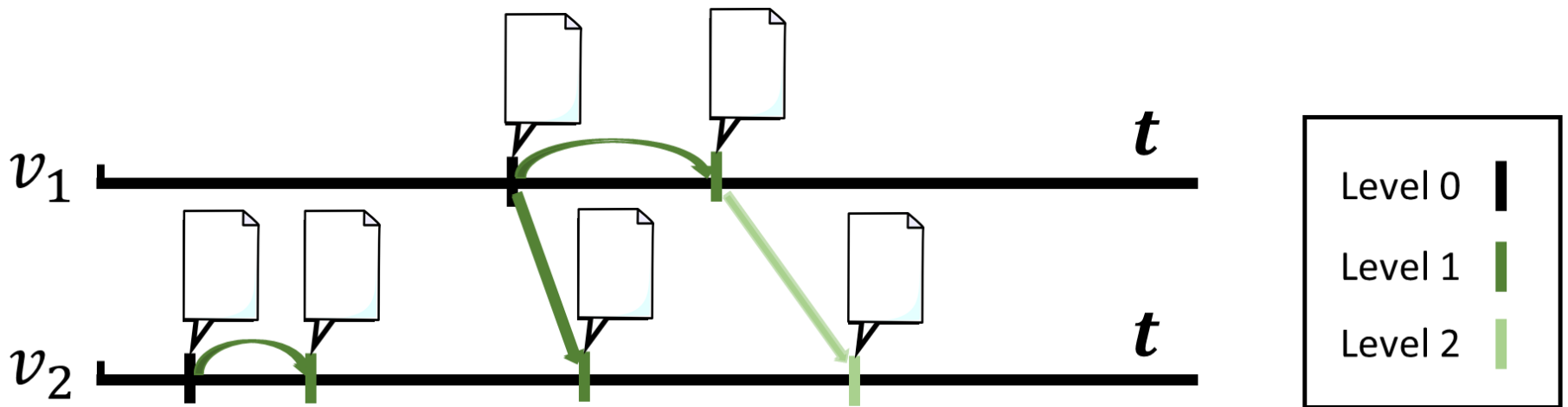
Rate **=** **Base intensity** **+** **Influence from previous events**

$$\lambda_v(t) = \underbrace{\mu_v}_{\text{Base intensity}} + \underbrace{\sum_{e:t_e < t} A_{v_e, v} f_{\Delta}(t - t_e)}_{\text{Influence from previous events}}$$

$A_{u, w}$: **influence strength** from u to v

$f_{\Delta}(\cdot)$: probability density function of the **delay distribution**

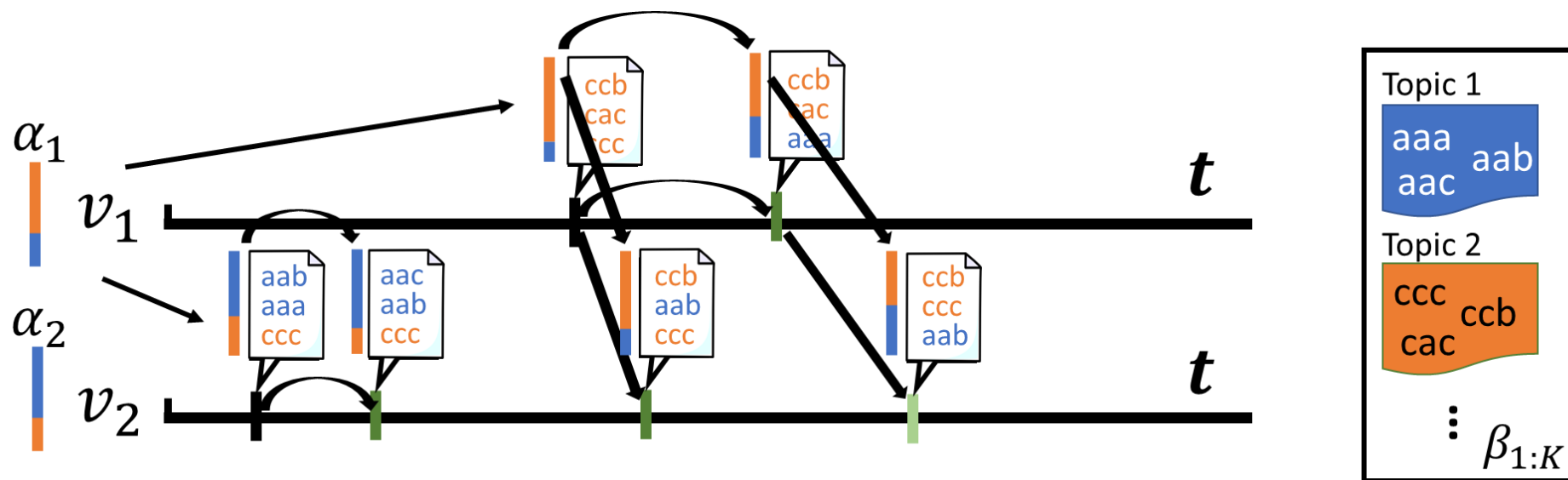
Clustered Poisson process interpretation



Generate events and their posting times in a **breadth first** order by interpreting the MHP as **clustered Poisson process** [Simma 10]

Provide explicit **parent relationship** for evolution of the content information

Generating documents



Step 1: Generate the topics $\beta_{1:K}: \beta_k \sim \text{Dir}(\alpha)$

Step 2: For spontaneous events (level=0): $\eta_e \sim N(\alpha_v, \sigma^2 I)$

Step 3: For triggered events (level>0): $\eta_e \sim N(\eta_{\text{parent}[e]}, \sigma^2 I)$

Step 4: For each word in each document: $z_{e,n} \sim \text{Discrete}(\pi(\eta_e)), x_{e,n} \sim \text{Discrete}(\beta_{z_{e,n}})$

Experiments for HawkesTopic



“Ebola” news articles ~4 months
~9k articles, 330 news media sites
Copying information as ground truth



High-energy physics theory papers ~12 years
Top 50/100/200 researchers
Citation network as ground truth

Evaluation metrics:

- Topic modeling: document competition likelihood [Wallach et al. 09]
- Network Inference: AUC against the ground truth network

Results: EventRegistry

Network Inference accuracy: **10%** improvement

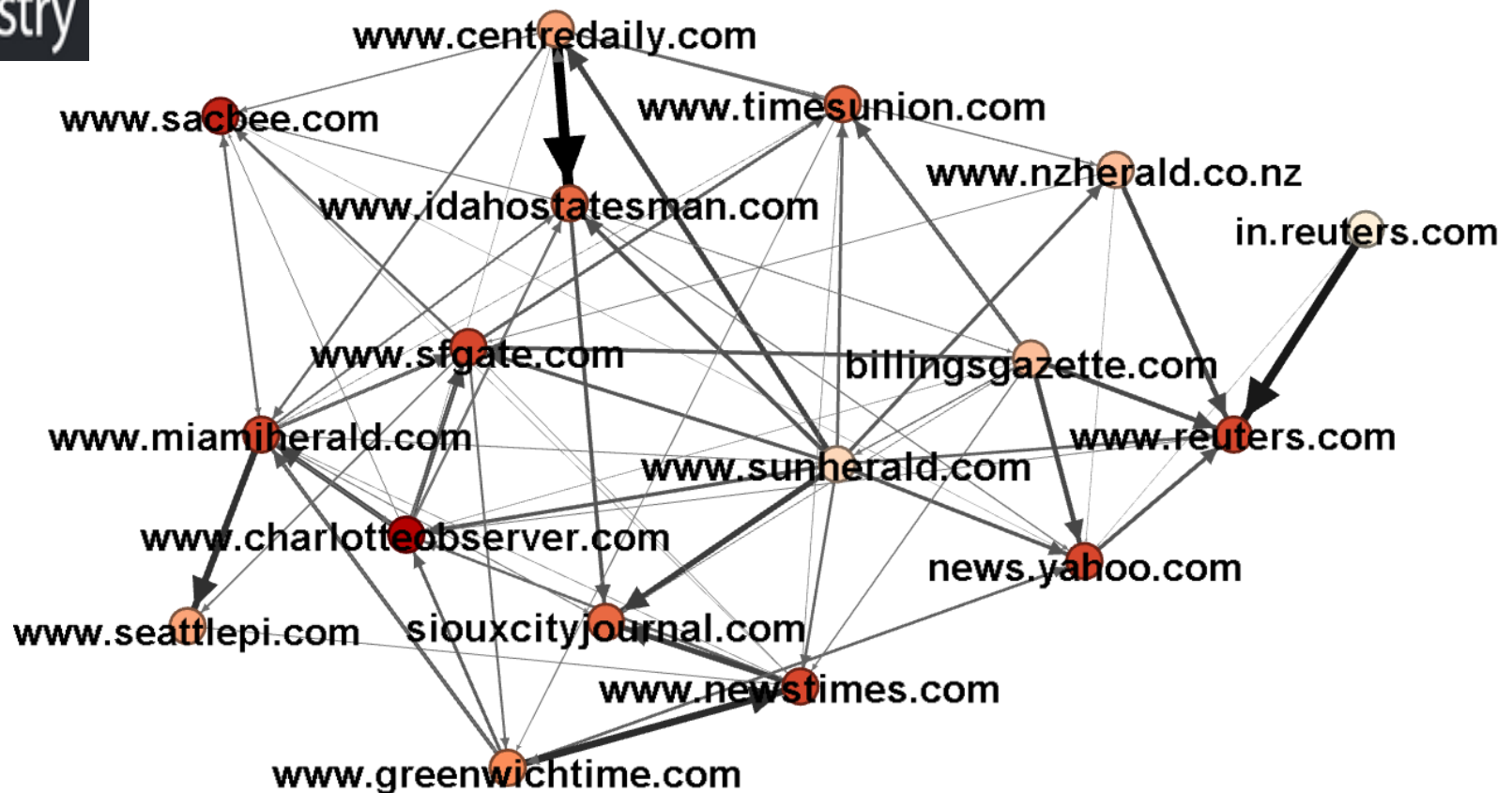
	Hawkes	Hawkes-LDA	Hawkes-CTM	HTM
Component 1	0.622	0.669	0.673	0.697
Component 2	0.670	0.704	0.716	0.730
Component 3	0.666	0.665	0.669	0.700

Topic modeling accuracy:

	LDA	CTM	HTM
Component 1	-42945	-42458	-42325
Component 2	-22558	-22181	-22164
Component 3	-17574	-17574	-17571

Results: EventRegistry

EventRegistry



Results: ArXiv

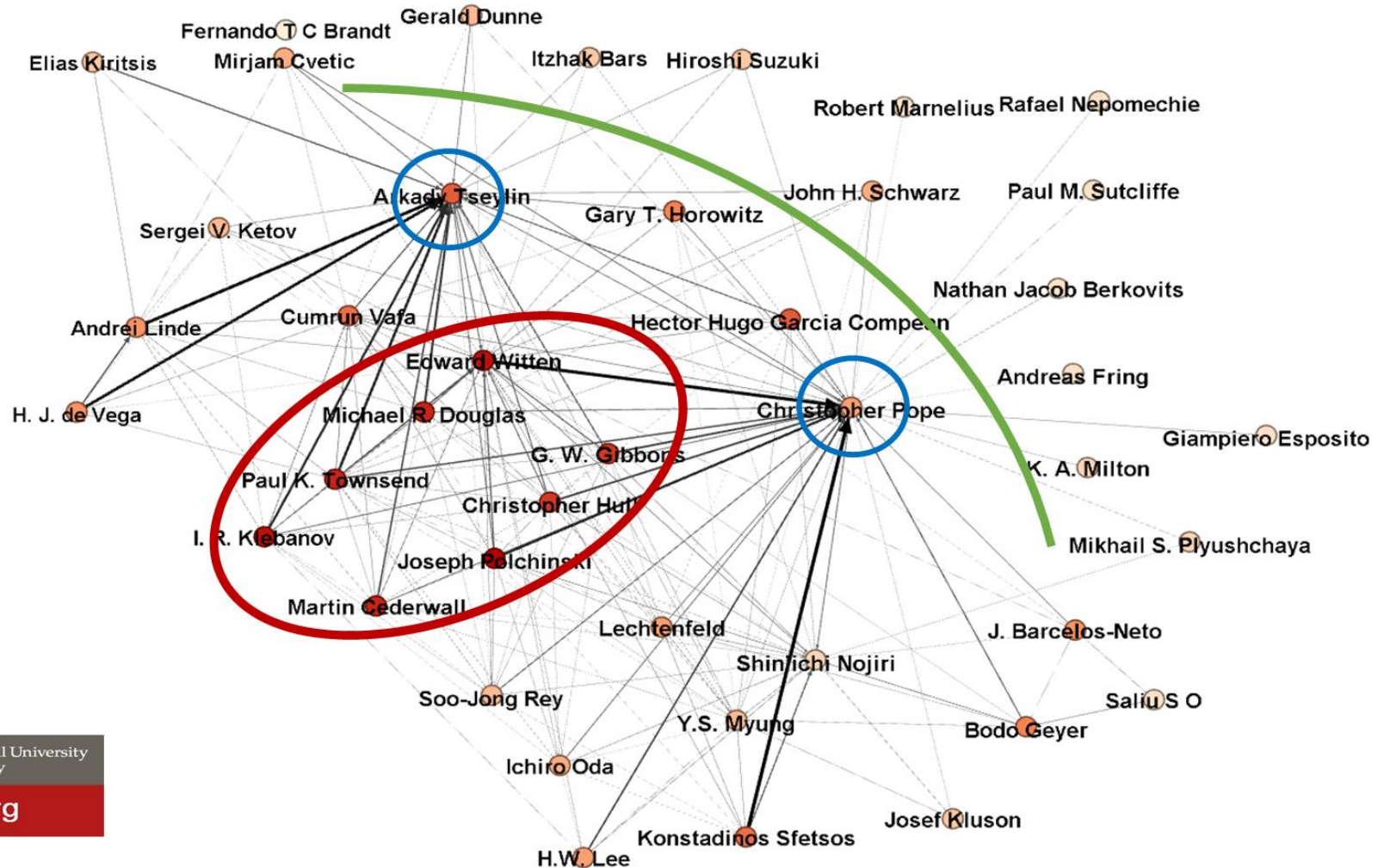
Network Inference accuracy: **40%** improvement

	Hawkes	Hawkes-LDA	Hawkes-CTM	HTM
Top50	0.594	0.656	0.645	0.807
Top100	0.588	0.589	0.614	0.687
Top200	0.618	0.630	0.629	0.659

Topic modeling accuracy:

	LDA	CTM	HTM
Top50	-11074	-10769	-10708
Top100	-15711	-15477	-15252
Top200	-27758	-27630	-27443

Results: ArXiv



Summary

- Generative models provide a powerful mechanism for modeling social networks
- Latent variable models offer flexible yet interpretable models motivated by sociological principles
 - Latent space model
 - Stochastic block model
 - Mixed-membership stochastic block model
 - Latent feature model
- Many recent advancements in generative models for social networks
 - Dynamic networks, cascades, joint modeling with text

Thank you!

The giant component

- Depending on the quantity $\lambda = pN$,
a “giant” connected component may emerge
- *P1)* If $\lambda < 1$, then a graph in $G(N, p)$ will have no connected components of size larger than $O(\log N)$, a.s. as $N \rightarrow \infty$.

The giant component

- Depending on the quantity $\lambda = pN$,
a “giant” connected component may emerge
- *P1*) If $\lambda < 1$, then a graph in $G(N, p)$ will have no connected components of size larger than $O(\log N)$, a.s. as $N \rightarrow \infty$.
- *P2*) If $\lambda = 1$, then a graph in $G(N, p)$ will have a largest component whose size is $O(N^{2/3})$, a.s. as $N \rightarrow \infty$.

The giant component

- Depending on the quantity $\lambda = pN$, a “giant” connected component may emerge
- *P1)* If $\lambda < 1$, then a graph in $G(N, p)$ will have no connected components of size larger than $O(\log N)$, a.s. as $N \rightarrow \infty$.
- *P2)* If $\lambda = 1$, then a graph in $G(N, p)$ will have a largest component whose size is $O(N^{2/3})$, a.s. as $N \rightarrow \infty$.
- *P3)* If λ tends to a constant $c > 1$, then a graph in $G(N, p)$ will have a unique “giant” component containing a positive fraction of the nodes, a.s. as $N \rightarrow \infty$.

No other component will contain more than $O(\log N)$ nodes, a.s. as $N \rightarrow \infty$.