Analysis of Social Media

MLD 10-802, LTI 11-772

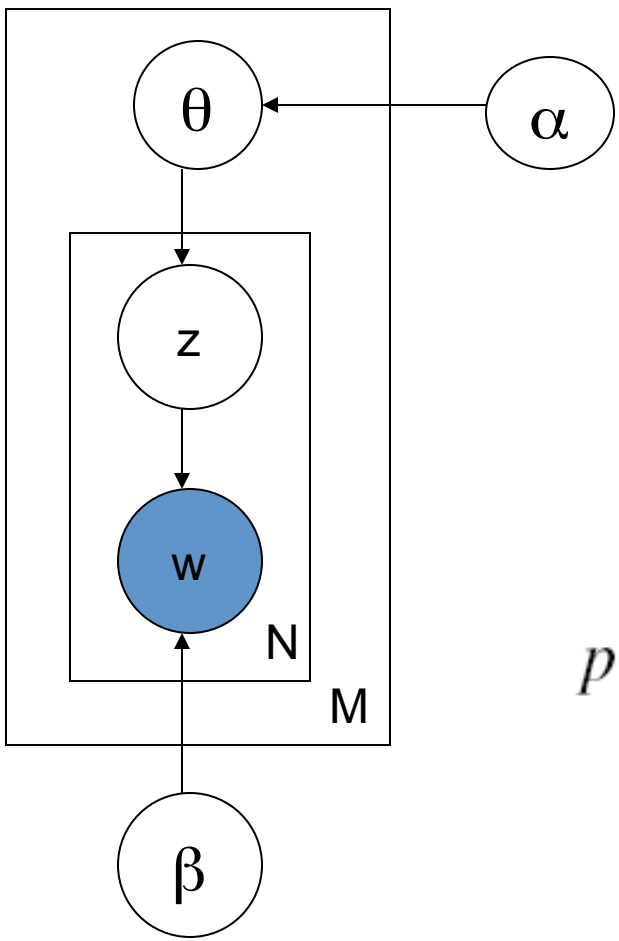
William Cohen

10-16-010

Review - LDA

"Mixed membership"

- Latent Dirichlet Allocation



- Randomly initialize each $z_{m,n}$
- Repeat for t=1,….
  - For each doc *m,* word *n*
    - Find Pr(z_{mn}=*k*|other z's)
    - Sample z_{mn} according to that distr.

$$p(z_i{=}k|\vec{z}_{\neg i}, \vec{w}) =$$

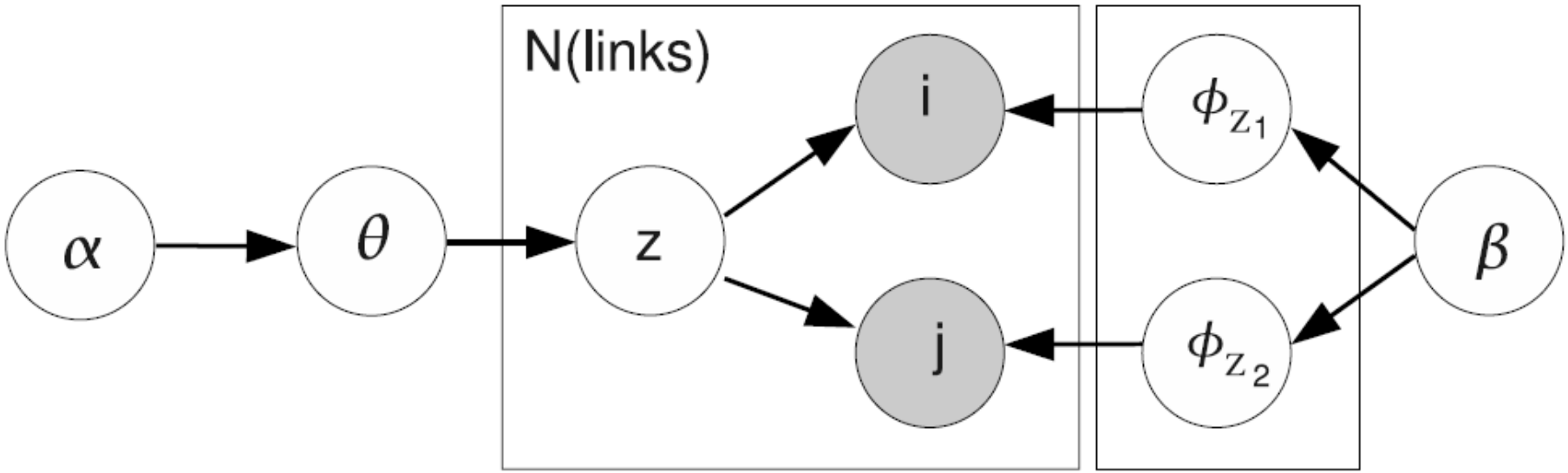
$$\propto \frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V} n^{(t)}_{k,\neg i} + \beta_t} \cdot \frac{n^{(t)}_{m,\neg i} + \alpha_k}{[\sum_{k=1}^{K} n^{(k)}_{m} + \alpha_k] - 1}$$

Outline

- Stochastic block models & inference question
- Review of text models
  - Mixture of multinomials & EM
  - LDA and Gibbs (or variational EM)
- **Block models and inference**
- Mixed-membership block models
- Multinomial block models and inference w/ Gibbs
- Beastiary of other probabilistic graph models
  - Latent-space models, exchangeable graphs, p1, ERGM

Parkkinen et al paper

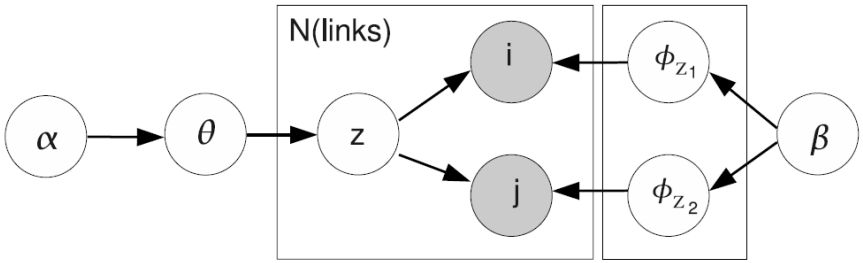
Another mixed membership block model



$$p(z_l|\{z\}^{\neg l}, \{(i,j)\}^{\neg l}, \alpha, \beta) \propto$$

$$(n_z^{\neg l} + \alpha) \cdot \frac{(q_{z_1 i}^{\neg l} + \beta)(q_{z_2 j}^{\neg l} + \beta)}{(q_{z_1 \cdot}^{\neg l} + M\beta)(q_{z_2 \cdot}^{\neg l} + M\beta + \delta_z)},$$

Another mixed membership block model



z=(zi,zj) is a pair of block ids

n_z = #pairs *z*

$q_{z1,i}$ = #links to *i* from block z1

$q_{z1,\cdot}$ = #outlinks in block z1

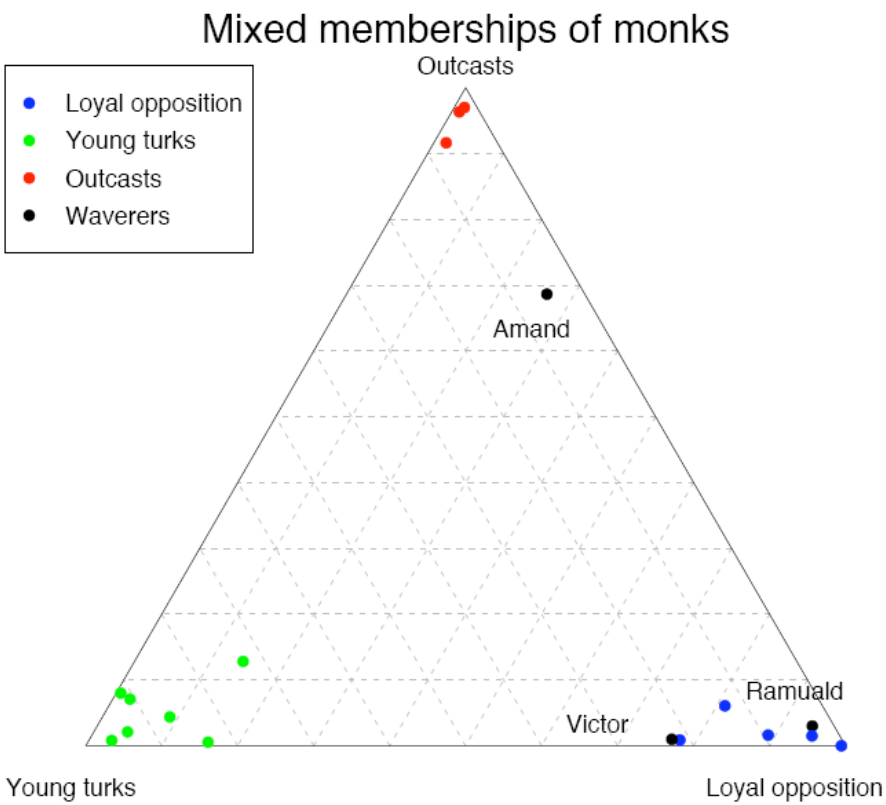
δ = indicator for diagonal

M = #nodes

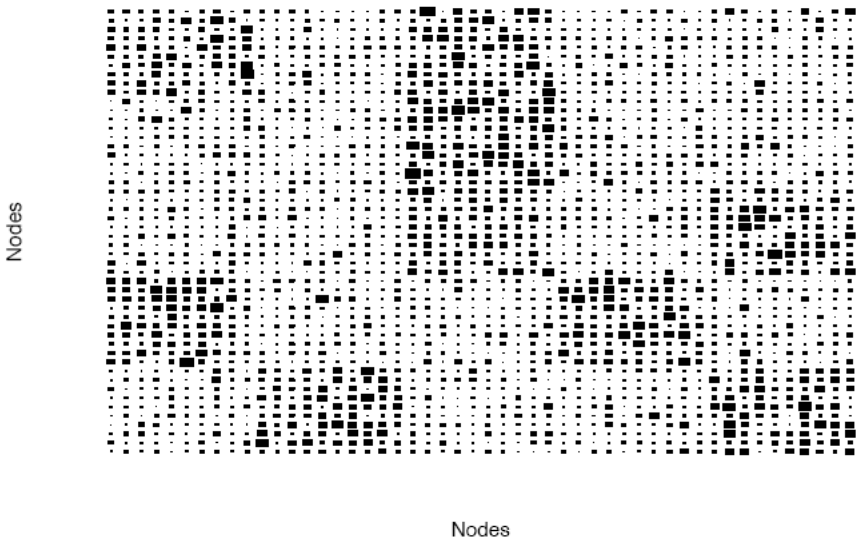
$$p(z_l|\{z\}^{\neg l}, \{(i,j)\}^{\neg l}, \alpha, \beta) \propto$$

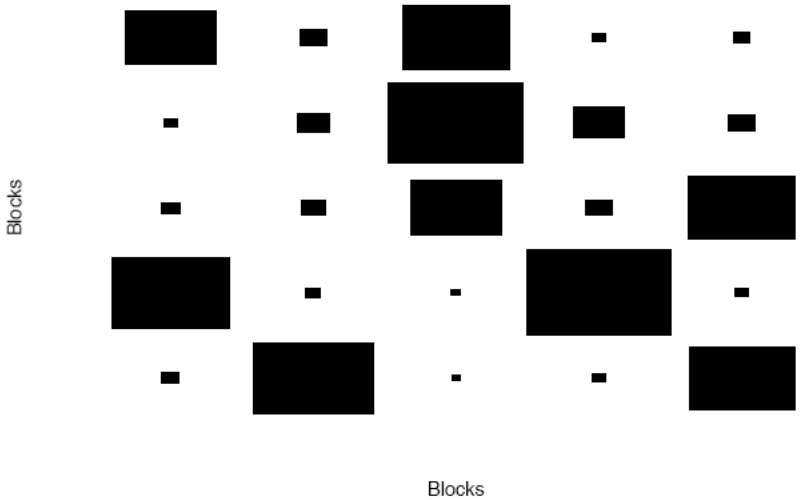
$$(n_z^{\neg l} + \alpha) \cdot \frac{(q_{z_1 i}^{\neg l} + \beta)(q_{z_2 j}^{\neg l} + \beta)}{(q_{z_1 \cdot}^{\neg l} + M\beta)(q_{z_2 \cdot}^{\neg l} + M\beta + \delta_z)},$$

Another mixed membership block model



Another mixed membership block model





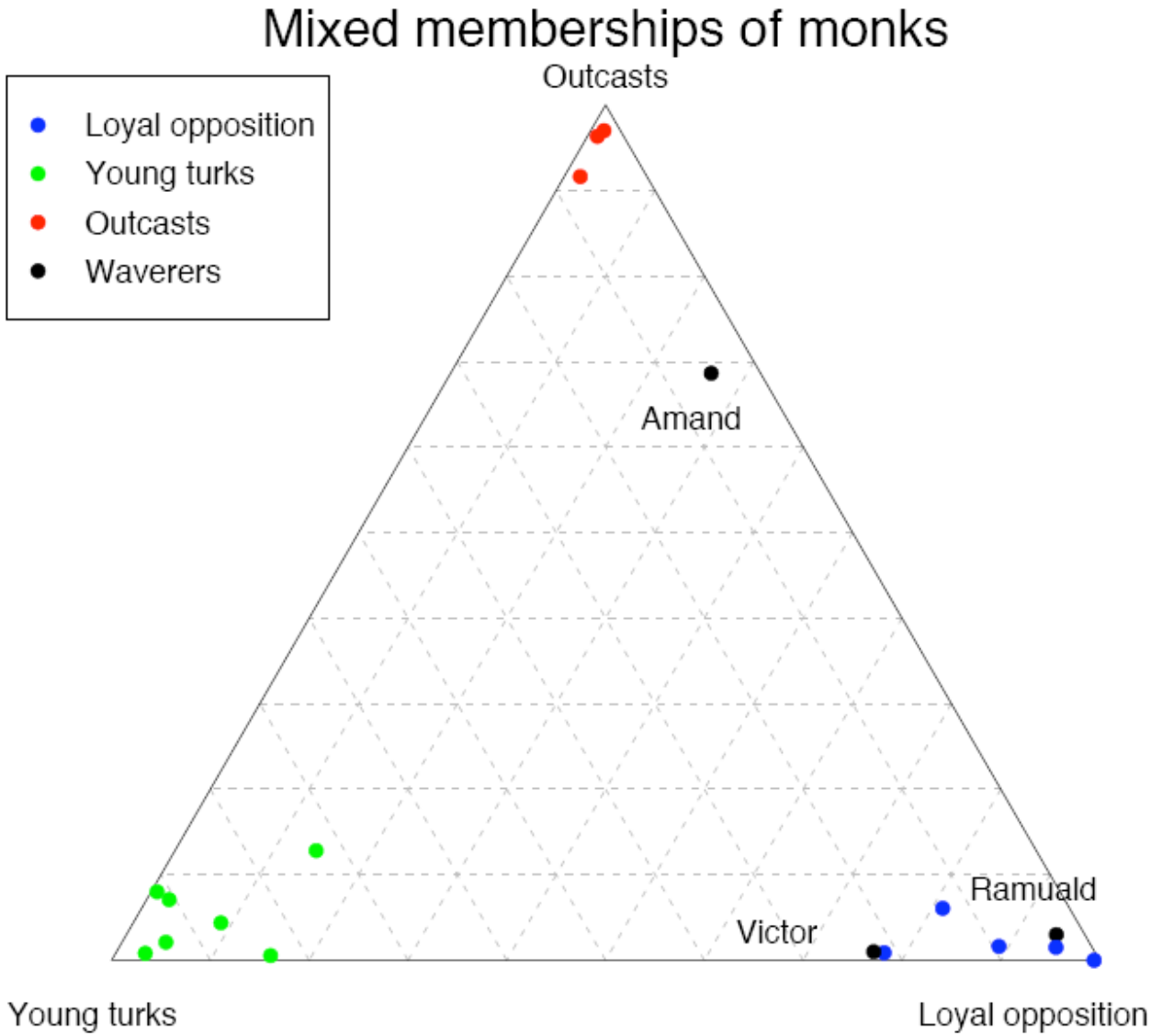
Outline

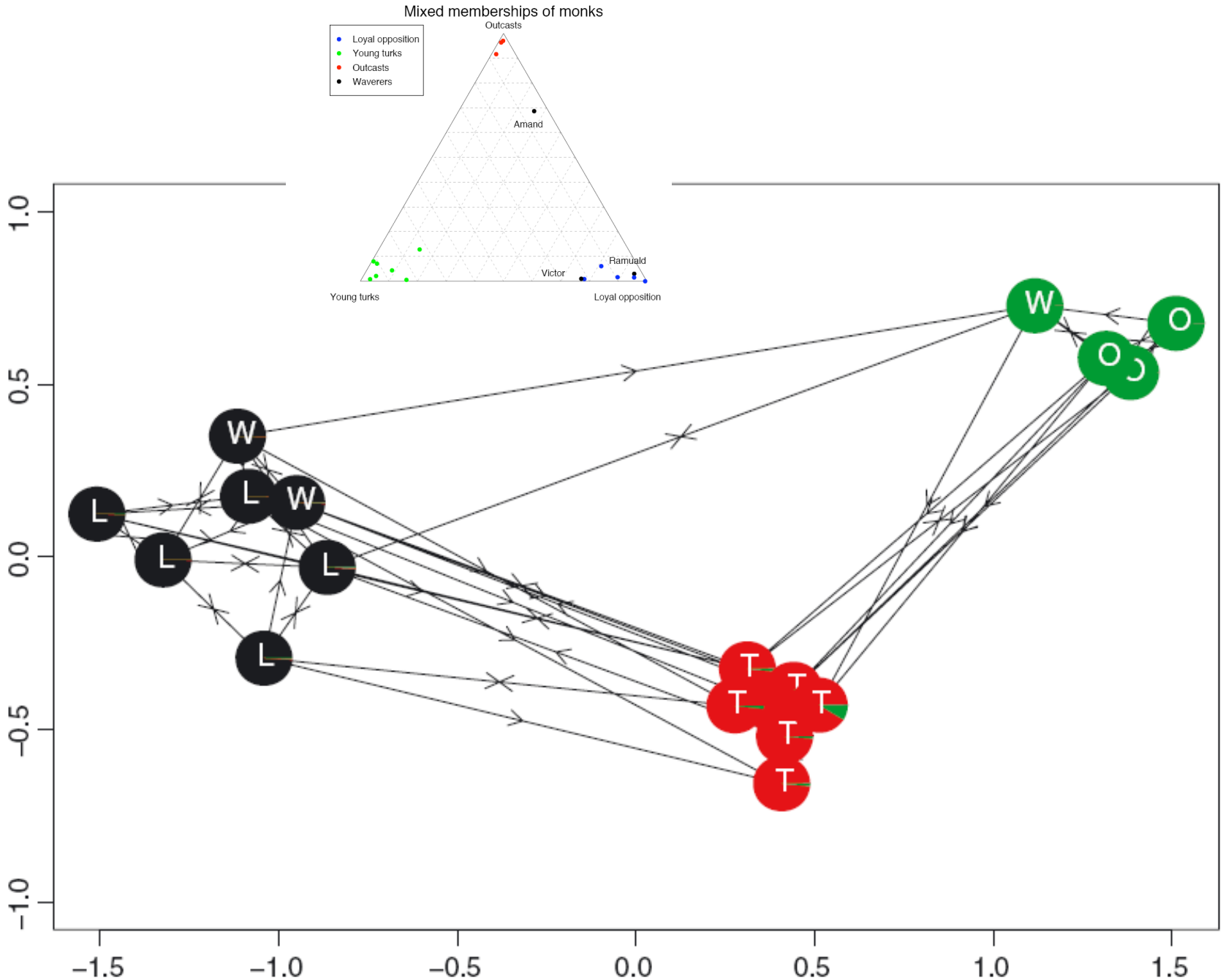
- Stochastic block models & inference question
- Review of text models
  - Mixture of multinomials & EM
  - LDA and Gibbs (or variational EM)
- Block models and inference
- Mixed-membership block models
- Multinomial block models and inference w/ Gibbs
- **Beastiary of other probabilistic graph models**
  - Latent-space models, exchangeable graphs, p1, ERGM

Latent Space Model

- Each node *i* has a latent position in Euclidean space, *z(i)*
- *z(i)'s* drawn from a mixture of Gaussians
- Probability of interaction between *i* and *j* depend on the *distance* between *z(i)* and *z(j)*
- Inference is a little more complicated… [Handcock & Raftery, 2007]

Airoldi's MMSBM




Mixed memberships of monks
Loyal opposition
Young turks
Outcasts
Waverers
Outcasts
Amand
Ramuald
Victor
Young turks
Loyal opposition
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
W
O
O
O
W
L
L
W
L
L
L
T
T
T
T
T
T

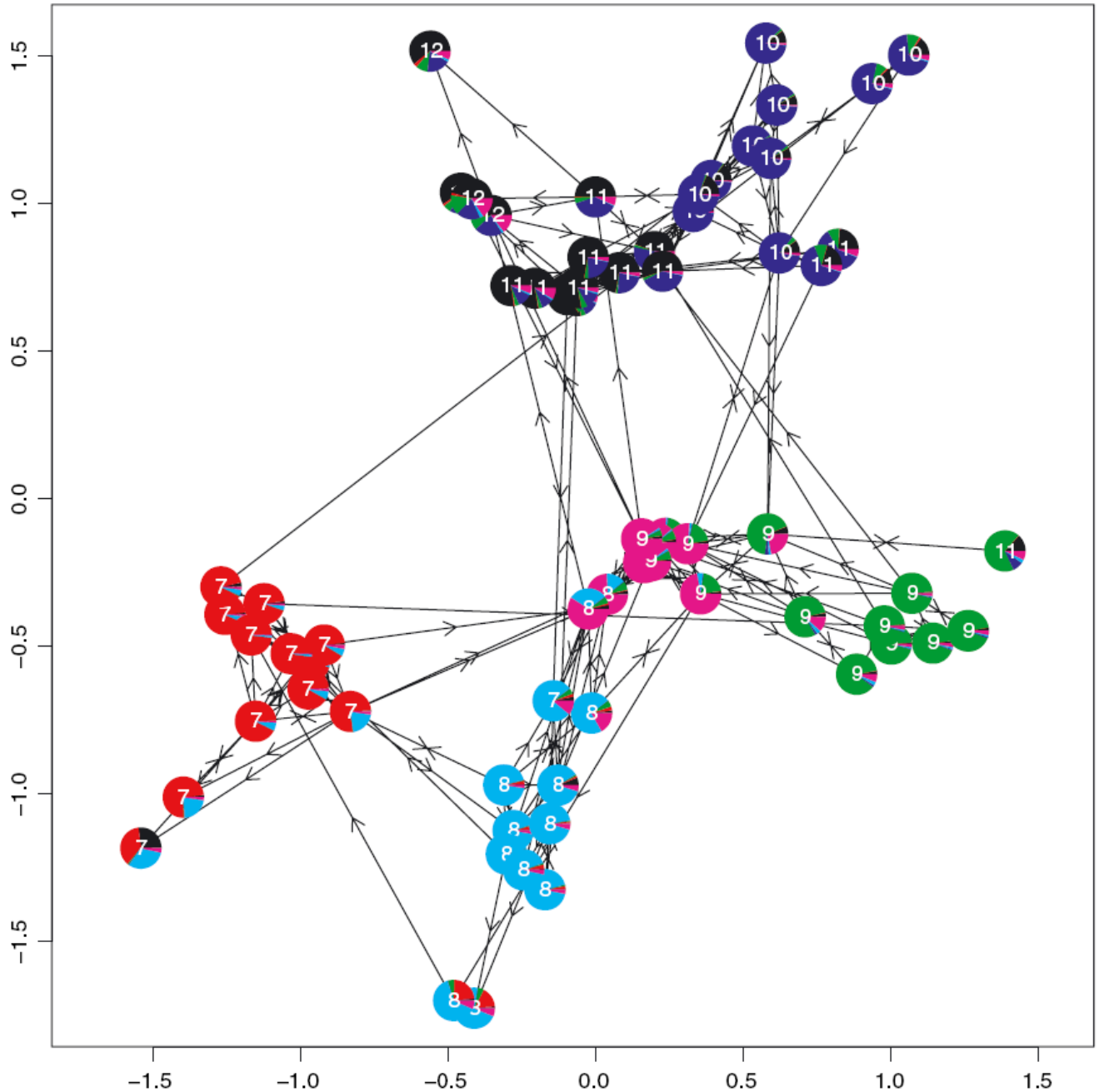


Fig. 8. Pie charts for posterior probabilities of cluster assignment for each actor, at the Bayesian estimates of posterior latent positions for the friendship network in the adolescent health school: the students' grades are shown as numbers

Outline

- Stochastic block models & inference question
- Review of text models
  - Mixture of multinomials & EM
  - LDA and Gibbs (or variational EM)
- Block models and inference
- Mixed-membership block models
- Multinomial block models and inference w/ Gibbs
- Beastiary of other probabilistic graph models
  - Latent-space models, exchangeable graphs, p1, ERGM

Exchangeable Graph Model

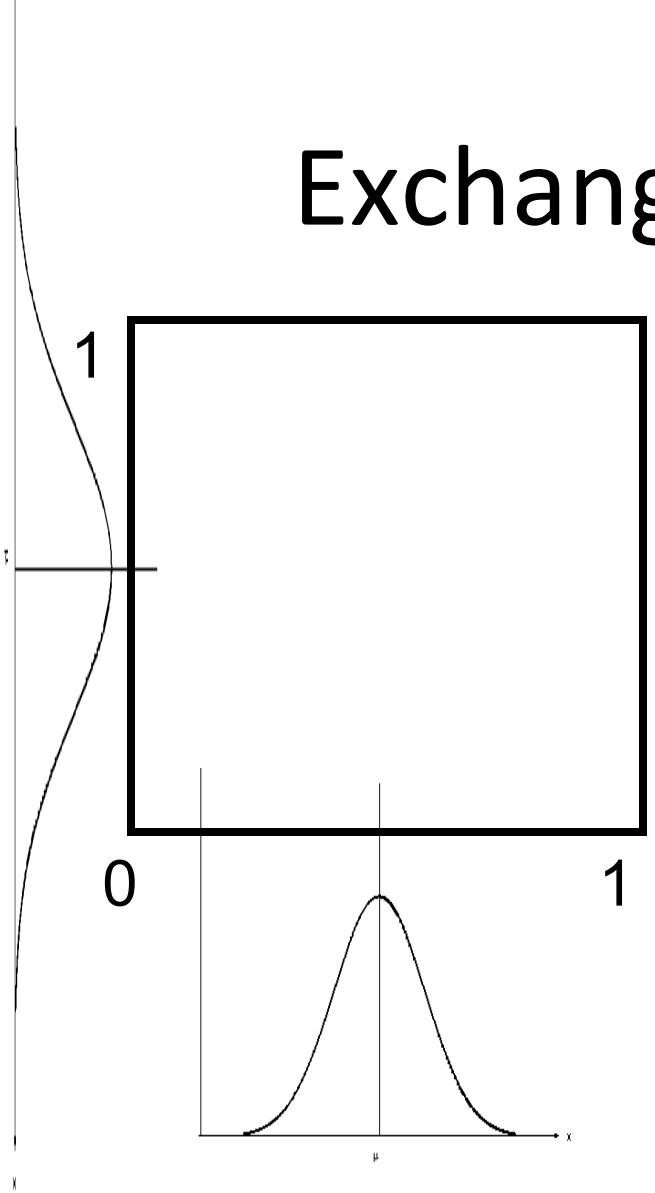
- Defined by a 2^k x 2^k table $q(b_1,b_2)$
- Draw a length-k bit string $b(n)$ like 01101 for each node n from a ~~*uniform*~~ distribution.

complicated

- For each pair o
  - Flip a coin wit
  - If it's heads co

- Pick k-dimensional vector u from a multivariate normal w/ variance α and covariance β – so u_i's are correlated.
- Pass each u_i thru a sigmoid so it's in [0,1] – call that p_i
- Pick b_i using p_i

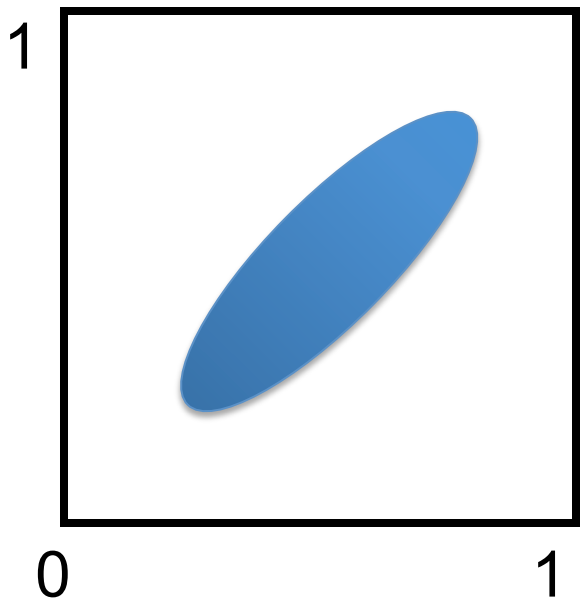
Exchangeable Graph Model



If α is big then *ux,uy* are really big (or small) so *px,py* will end up in a corner.

- Pick *k*-dimensional vector *u* from a multivariate normal w/ variance α and covariance β – so u_i's are correlated.
- Pass each u_i thru a sigmoid so it's in [0,1] – call that p_i
- Pick b_i using p_i

Exchangeable Graph Model



If α is big then *ux,uy* are really big (or small) so *px,py* will end up in a corner.

- Pick *k*-dimensional vector *u* from a multivariate normal w/ variance α and covariance β – so u_i's are **correlated**.
- Pass each u_i thru a sigmoid so it's in [0,1] – call that p_i
- Pick b_i using p_i

The p_1 model for a directed graph

- Parameters, per node *i:*
  - Θ: background edge probability
  - α_i: "expansiveness" – how extroverted is *i?*
  - β_i: "popularity" – how much do others want to be with *i?*
  - ρ_{ij}: "reciprocation" – how likely is *i* to respond to an incomping link with an outgoing one?

$$\log \Pr(i \ldots j) = \lambda_{ij}$$

$$\log \Pr(i \rightarrow j) = \lambda_{ij} + \boxed{\alpha_i + \beta_j + \theta}$$

$$\log \Pr(i \leftarrow j) = \lambda_{ij} + \boxed{\alpha_j + \beta_i + \theta}$$

$$\log \Pr(i \leftrightarrow j) = \lambda_{ij} + \boxed{} + \boxed{} + \rho_{ij}$$

Logistic-regression like procedure can be used to fit this to data from a graph

$$\log Pr_{p_1}(y) \propto y_{++}\theta + \sum_i y_{i+}\alpha_i + \sum_j y_{+j}\beta_j + \sum_{ij} y_{ij}y_{ji}\rho,$$

Exponential Random Graph Model

- Basic idea:
  - Define some features of the graph (e.g., number of edges, number of triangles, …)
  - Build a MaxEnt-style model based on these features
- General:
  - includes Erdos-Renyi, p_1, …
- Issues
  - Partition function is intractible
  - Alternative: model conditional pseudo-likelihood of a each edge (i.e., Pr(edge|rest of graph)

Kroneker product graphs

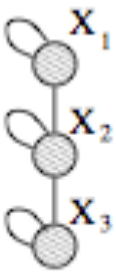
Jure Leskovec
Computer Science Department, Stanford University

Deepayan Chakrabarti
Yahoo! Research

Jon Kleinberg
Computer Science Department, Cornell University

Christos Faloutsos
Computer Science Department, Carnegie Mellon University

Zoubin Ghahramani
Department of Engineering, University of Cambridge
Machine Learning Department, Carnegie Mellon University



(a) Graph K_1

1	1	0
1	1	1
0	1	1

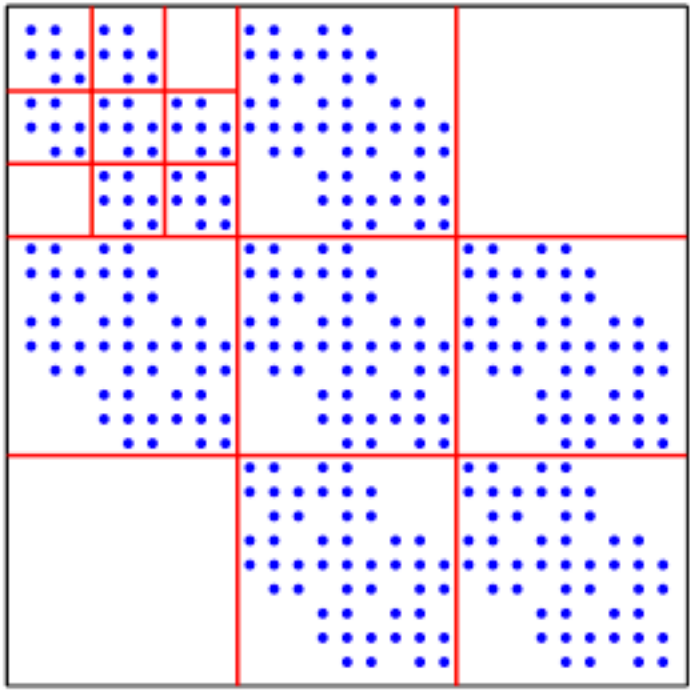
(d) Adjacency matrix of K_1

Kroneker product graphs

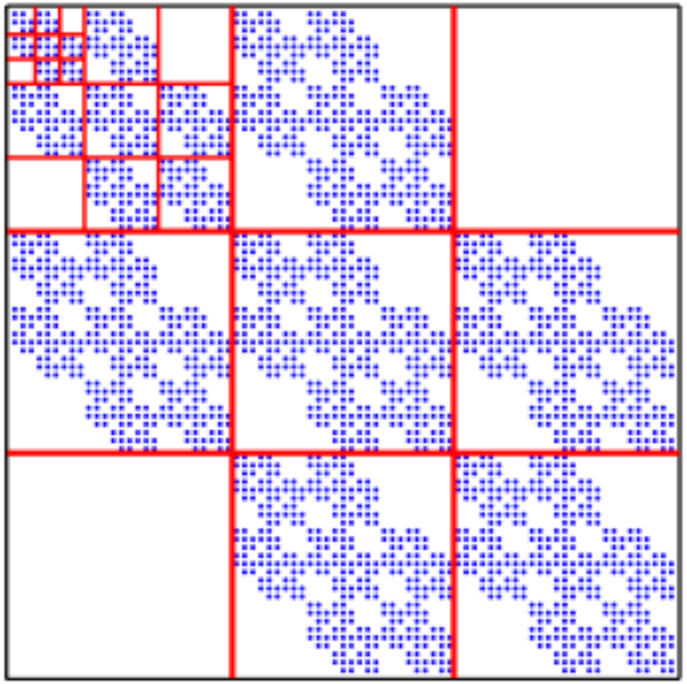
(a) Graph K_1

1	1	0
1	1	1
0	1	1

(d) Adjacency matrix of K_1



(a) K_3 adjacency matrix (27×27)



(b) K_4 adjacency matrix (81×81)

Kroneker product graphs

- Good fit to many commonly-observed network properties
  - scale-free degree distribution
  - diameter
  - …
- Gradient descent can be used to fit an "initiator matrix" to a real adjacency matrix