Inferring vertex properties from topology in large networks

Janne Aukia
(Xtract Ltd)

with:

Janne Sinkkonen
(Xtract Ltd)

Samuel Kaski
(Helsinki University of Technology)

Contents

- Overview of the problem
- Background
  - Generative modeling
- Latent component model
  - Infinite components
  - Sampling
- Results

Interactions as networks

- Many types of interactions can be represented as large networks
  - Friendships between people, protein interactions, web pages...
  - Missing data and imprecise relationships
  - Nodes and edges are often unlabeled
- Networks have often some type of structure
  - Dense groups of nodes
  - Number of links between nodes (degree) varies

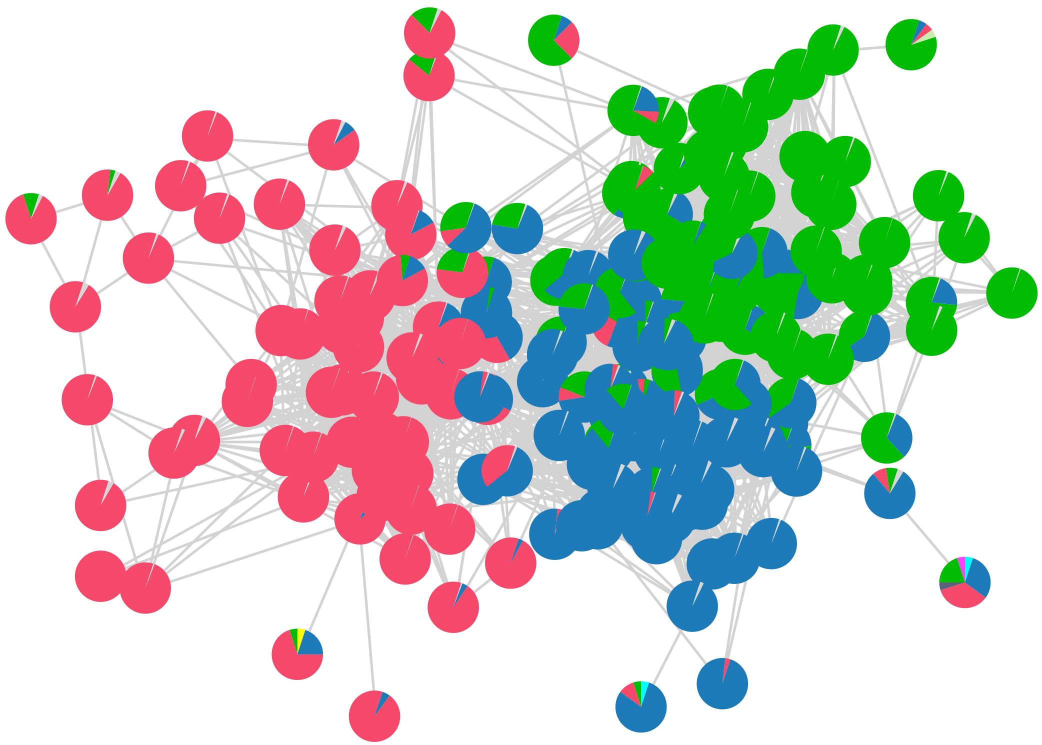
Problem setting

- How to find the underlying factors which can explain network structure for a single, unlabeled, large graph?
- Some previous approaches
  - Community detection (Newman & Girvan 2004)
  - Machine learning (Airoldi et al. 2006, Handcock et al. 2007)
- Our approach
  - A latent component model
  - Generative model for constructing edges in graphs
  - Optimized with collapsed Gibbs sampling
  - Usable on networks with millions of nodes

[1] Airoldi E. M., Blei D. M., Fienberg S. E., Xing E. P. (2006). Mixed-membership stochastic block models for relational data with application to protein-protein interaction.
[2] Handcock M. S. and Raftery A. E. (2007). Model-based clustering for social networks. J. R. Statist. Soc. A 170, 1–22.
[3] Newman, M. E.J. and Girvan, M. (2004). Finding and evaluating community structure in networks. Physical Review E, 69:026113.

Example of structure

- A collaboration network of jazz musicians[1] has community structure

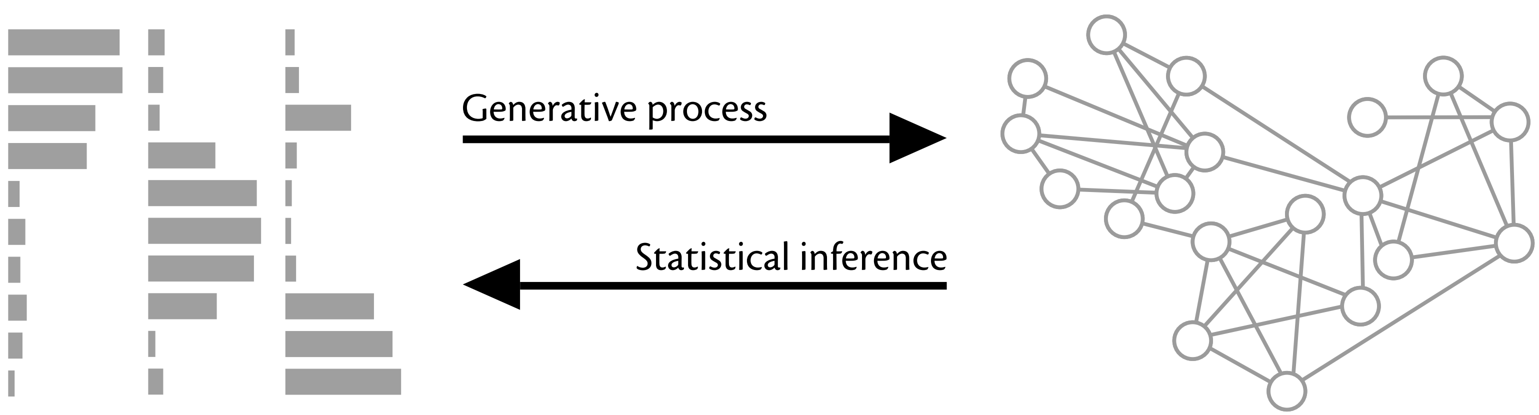


Components found with
the latent component algorithm

[1] P. Gleiser and L. Danon, Adv. Complex Syst. 6, 565 (2003). Data at: http://deim.urv.cat/~aarenas/data/welcome.htm

Generative modeling

- A generative model can generate samples of the data it represents from a set of parameters
  - “Cooking recipe”
- Models are often hierarchical
- Bayesian methods can be used to infer model parameters from a sample



Latent component model

- Each node belongs to a number of latent components
  - Mixture of components
- Generative model, for each edge:
  - A component is selected based on the component probabilities
  - Edge endpoints are selected based on the probability of the endpoint in the component
- Probabilities for components and nodes in components are drawn from Dirichlet distributions

Illustration of the model

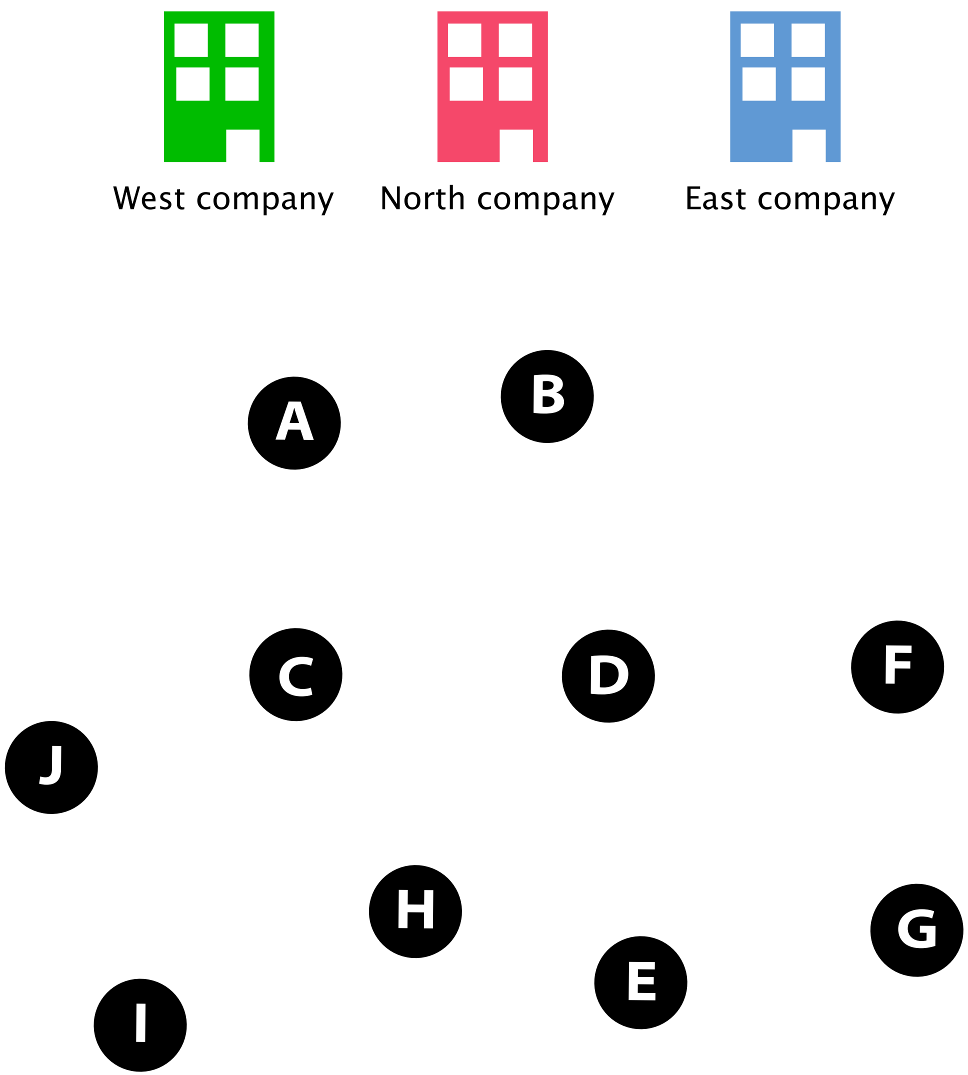


Illustration of the model

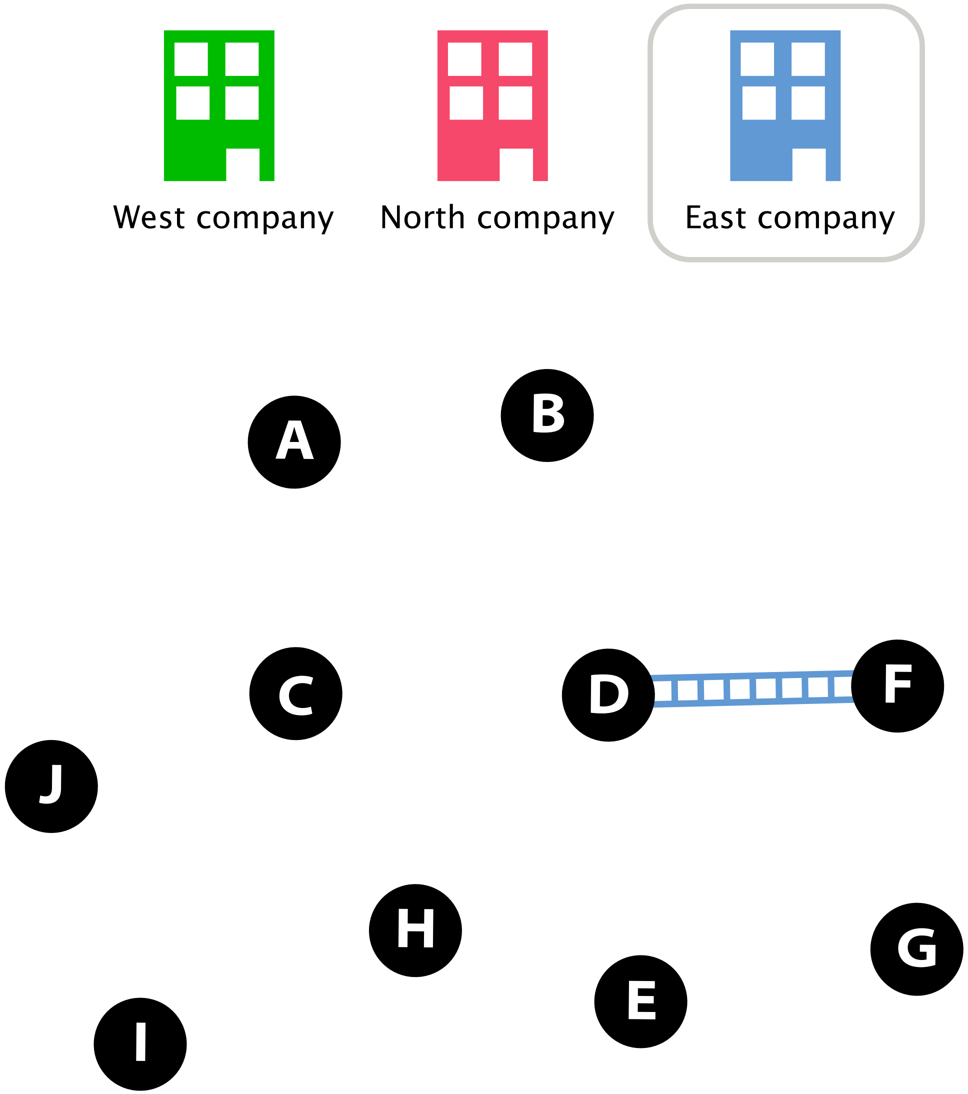


Illustration of the model

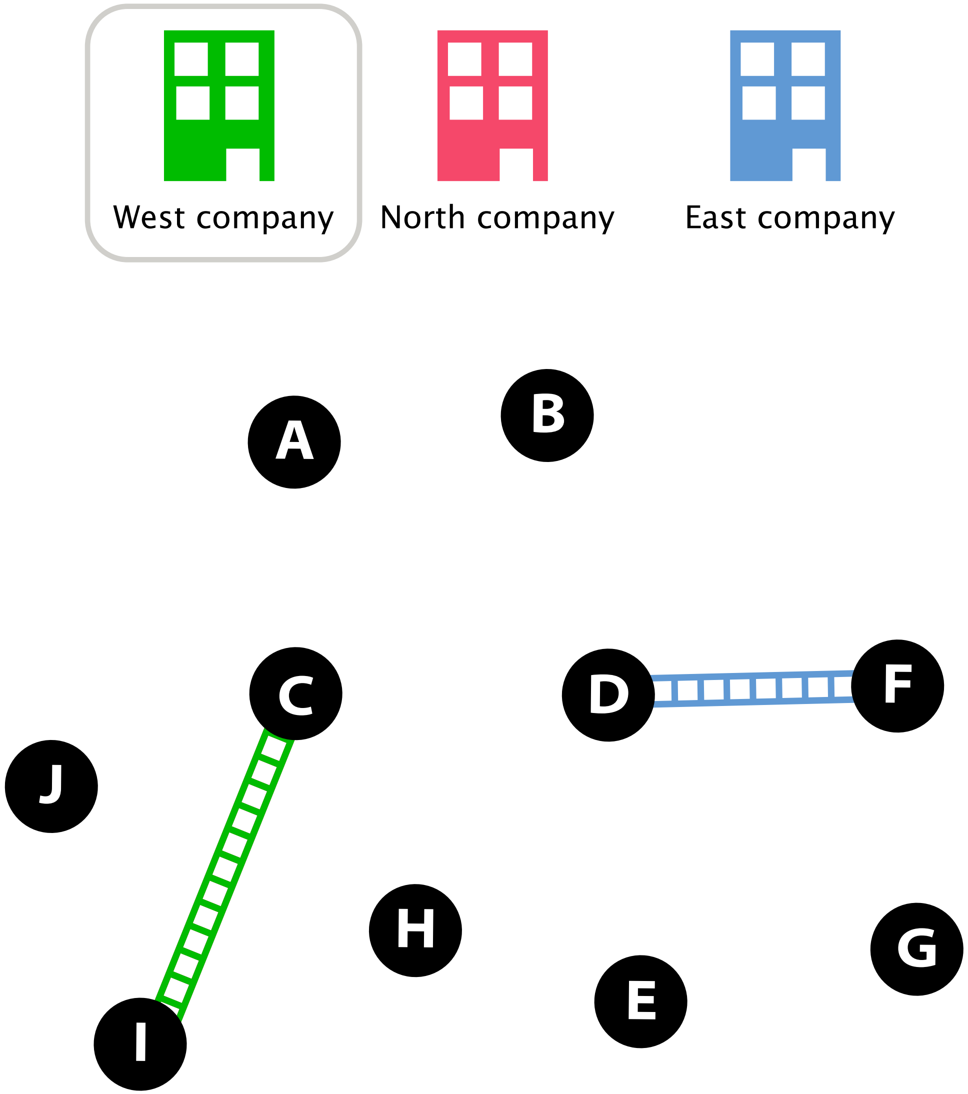


Illustration of the model

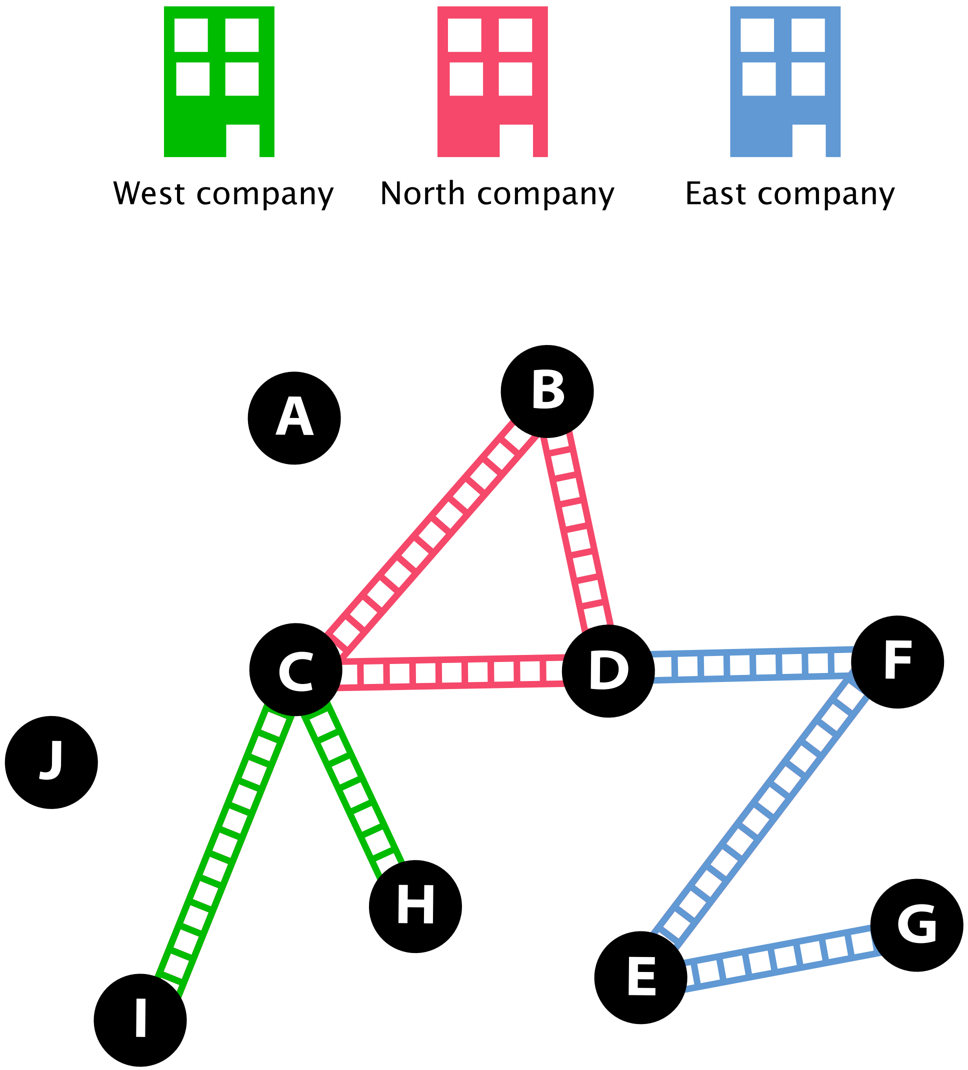
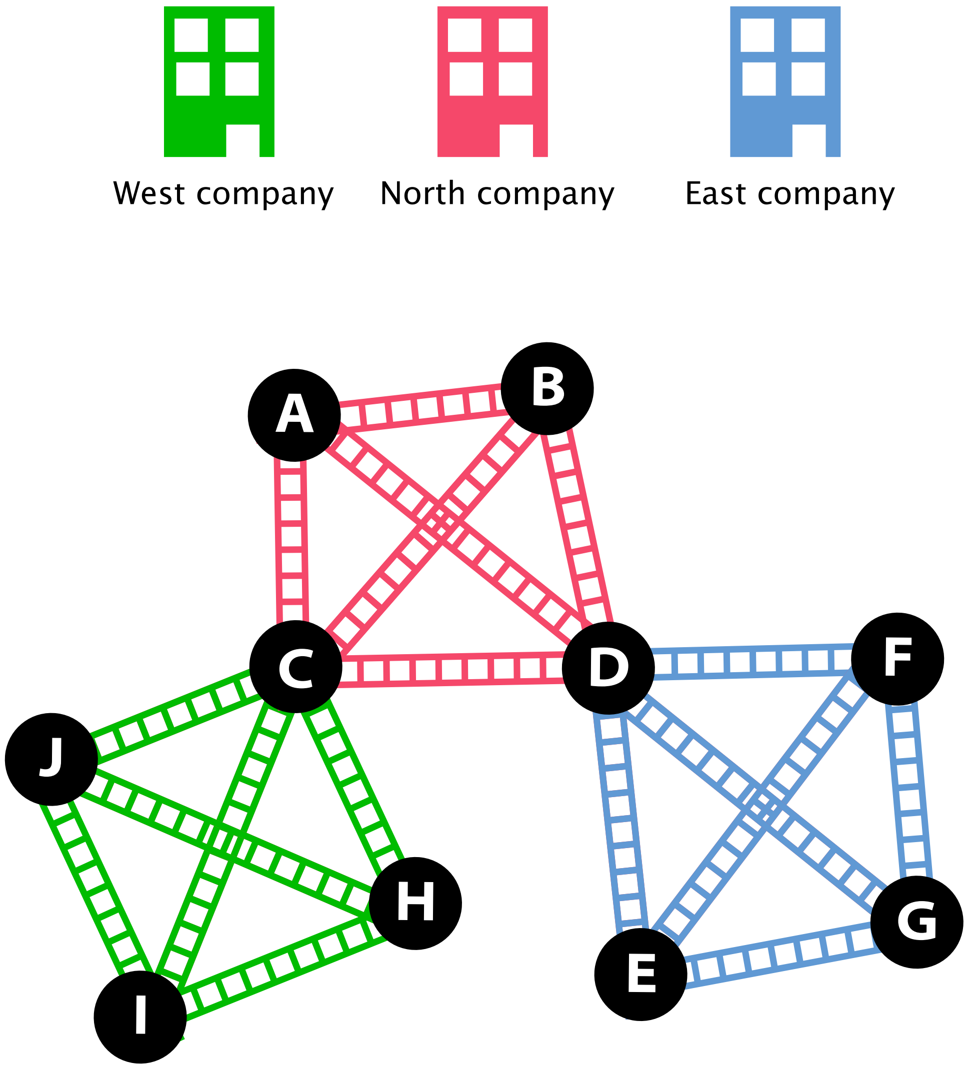
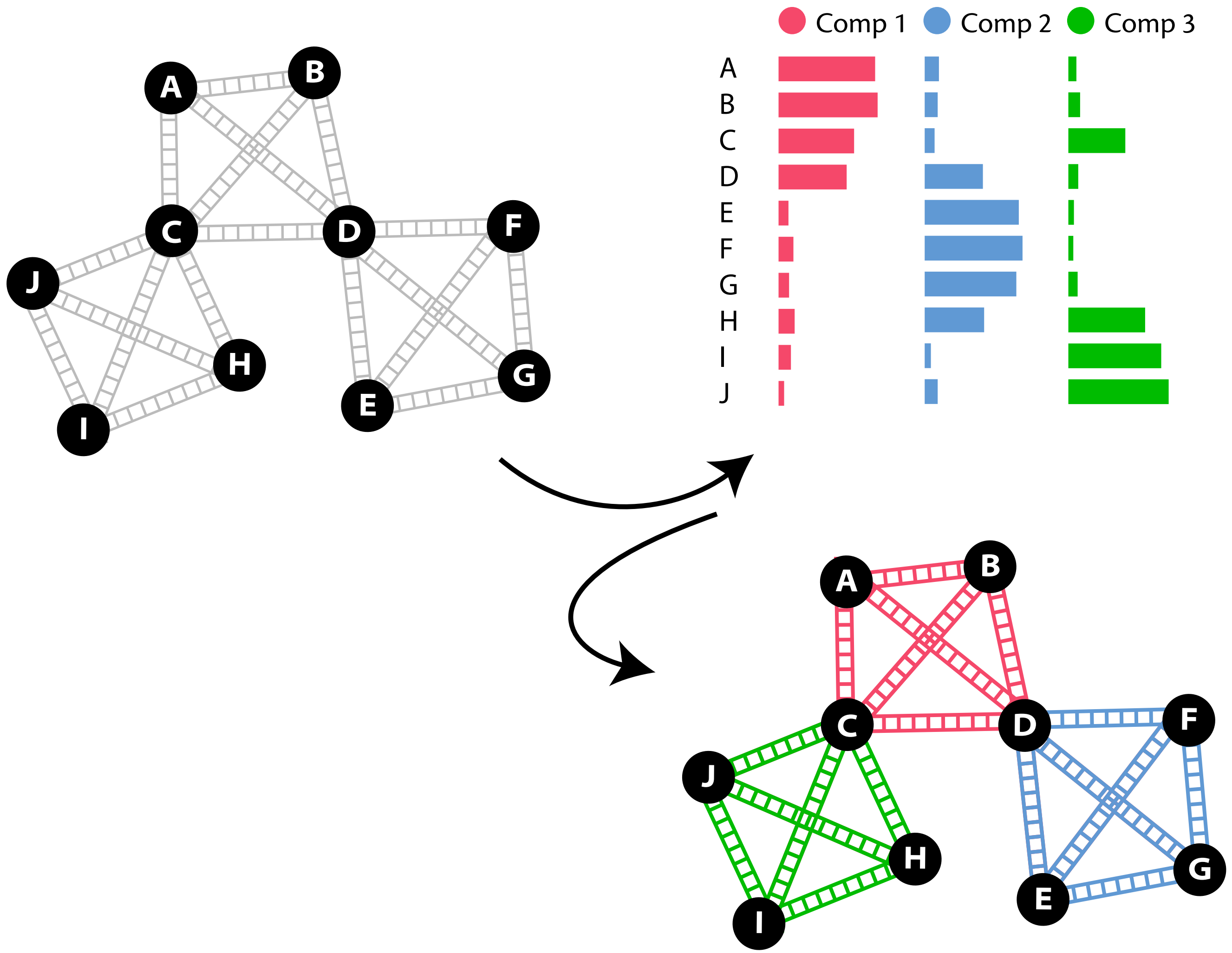


Illustration of the model



Parameter inference



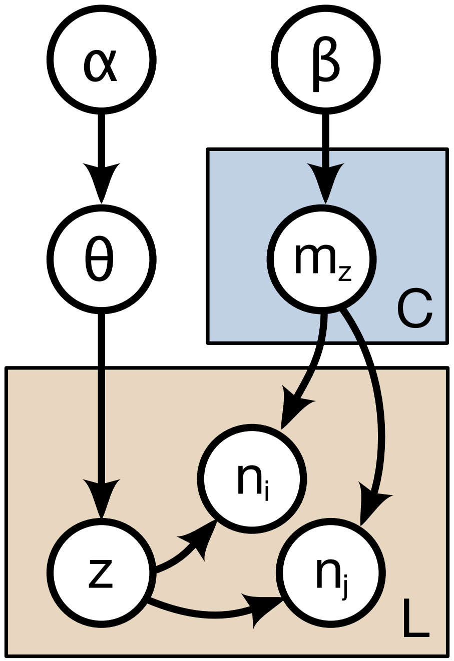
Infinite mixture

- A crucial feature in latent component models is to learn the number of components required
  - Can be achieved by using a Dirichlet process (DP)
- DP corresponds to Dirichlet distribution with infinite components
  - In practice, leads to a finite number of components
- Estimates the amount of components from data
  - However, hyperparameter (α) remains

Generative process

- Full generative process for the infinite component model:

1. Draw θ from $DP(\alpha)$
2. For each component z in C components:
   (a) Draw m_z from $Dir(\beta)$
3. For each of L edges:
   (a) Draw a latent component z from θ
   (b) Draw first end point n_i from m_z
   (c) Draw second end point n_j form m_z



Inferring components

- From the full model and its joint distribution, latent components can be found using Bayesian inference
  - A form of unsupervised learning
- Because of the Dirichlet priors, the inference is tractable and can be easy to compute
- Components can be found with EM optimization or full MCMC inference
  - EM seems to converge to bad local minima
  - Gibbs sampling, a form of MCMC, gives better results
- An effective implementation with collapsed Gibbs sampling
  - Latent variables marginalized away, only counts remain!

Joint distribution

- The joint probability distribution for the infinite mixture model:

$$p_{DP}(L, Z, m|\alpha, \beta) = p(L|Z, m) \times p(m|\beta) \times p(Z|\alpha)$$
$$= \prod_{iz} m_{zi}^{k_{zi}} \times \frac{\prod_{iz} m_{zi}^{\beta-1}}{D(E, \beta)^C} \times \frac{2E!\alpha^C}{C!\alpha_{2N} \prod_z n_z}$$

$$\alpha_{2N} = \alpha(\alpha + 1) \ldots (\alpha + 2N - 1).$$

Conditional probability

- Sampling implemented with Gibbs sampler
- Conditional probability for each edge conditioned on all the other edges
  - Unknown parameters marginalized away
- Component probabilities for the left out edge:

$$p(z|i,j) = \frac{k_{zi} + \beta}{2n_z + 1 + M\beta} \times \frac{k_{zj} + \beta}{2n_z + M\beta} \times \frac{C(n_z, \alpha)}{N + K\alpha}$$

$$C(n_z, \alpha) = n_z \text{ if } n_z \neq 0 \text{ and } \boxed{C(0, \alpha) = \alpha}$$

New component

- In every iteration, a component is sampled for each edge based on the conditional probabilities

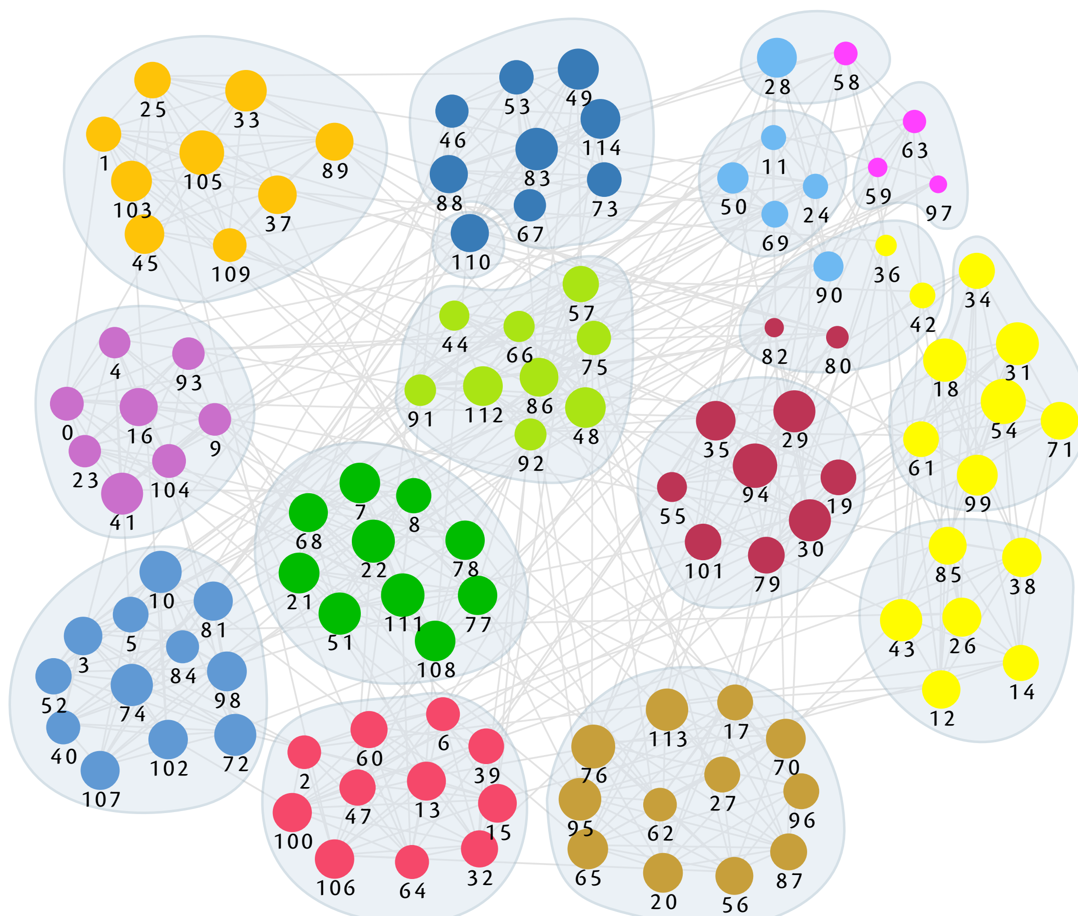
Example 1: Football network

- The football network[1] depicts American college football games during fall season 2000
  - 115 nodes (teams) and 613 edges (games)
  - A standard test data for clustering networks
  - Known community structure (clustering), teams belong to different conferences

[1] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002).
Data at: http://www-personal.umich.edu/~mejn/netdata/

Football result

- Colors represent clusters
- Blue background represents the correct clustering into conferences

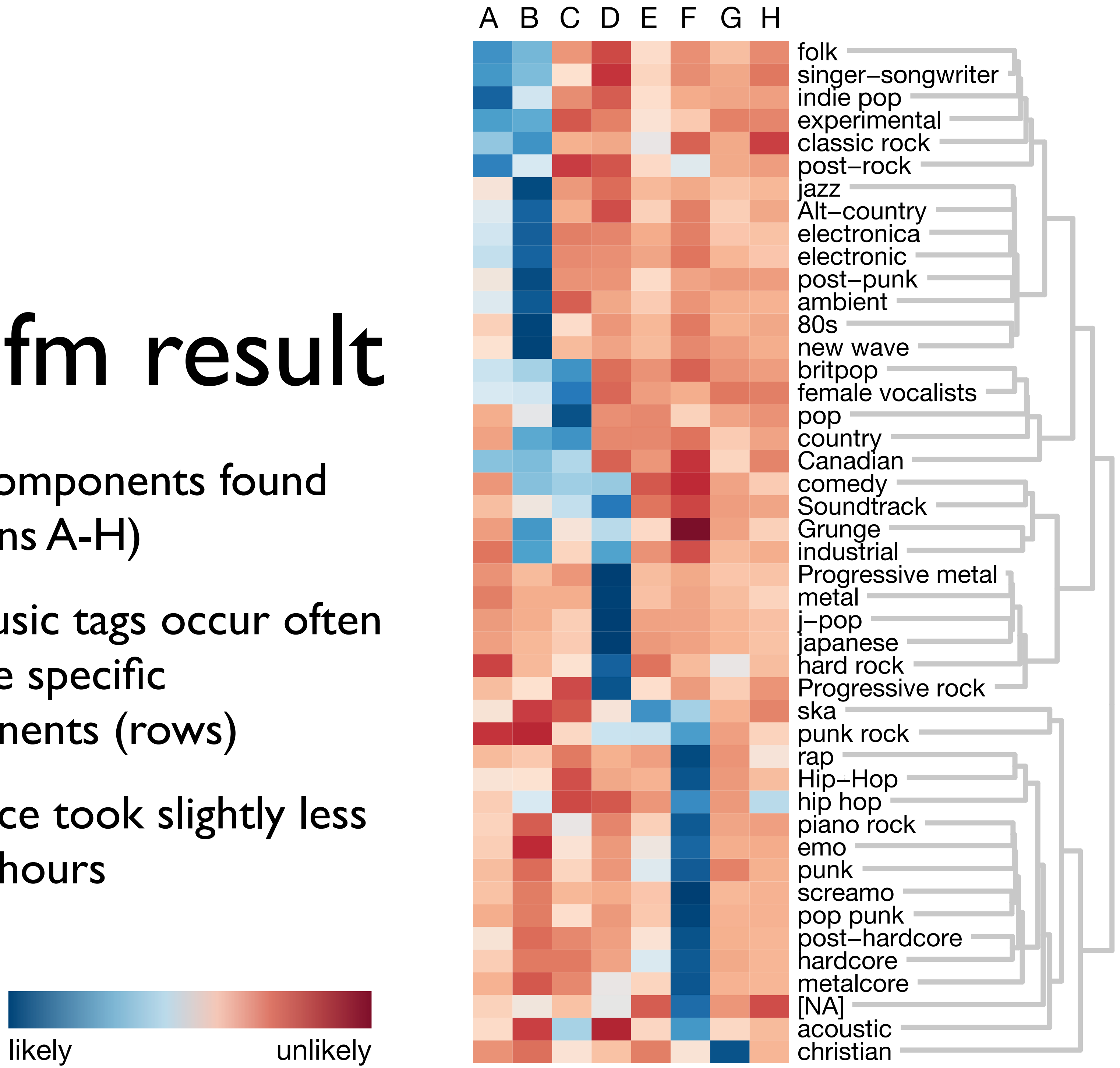


Example 2: Last.fm

- A large friendship network of 675,681 Last.fm users
  - Crawled via Last.fm web services during March and April 2007
  - Mutual links between all users
  - Subset: 147,610 users claiming to be from the US
- For each user: demographics (age, country, sex) and music taste (artists)
- In addition, tags for over 188,565 artists were crawled

Last.fm result

- Eight components found (columns A-H)
- The music tags occur often in some specific components (rows)
- Inference took slightly less than 4 hours



Conclusion

- Algorithm performs well at clustering networks
  - Can find both local structure (clusters) and diffuse global traits (latent dimensions)
  - Method is computationally efficient
    - However, suboptimal hierarchical clustering methods are even faster
  - Provides information on the confidence of the clustering results
- Choice of constant parameters for the model (hyperparameters) may be hard

Future work

1. Further validation of algorithm
   - Perform comparisons with machine learning methods and community extraction algorithms
   - More detailed analysis of the algorithm as a predictor for node traits
2. Method development
   - Include more information about network structure into the model, such as weights, user traits, directed links
   - Model architecture
   - Distributional assumptions
3. Improvement of performance
   - Parallel implementation of sampling