Inferring vertex properties from topology in large networks

Janne Aukia*, Juuso Parkkinen†, Samuel Kaski†, and Janne Sinkkonen*

* Xtract Ltd, Helsinki, Finland ,
† Adaptive Informatics Research Centre and Helsinki Institute for Information Technology, Helsinki University of Technology, Finland

Introduction

Large networks can be used to represent many types of interactions, such as friendships between people, protein interactions, and linked web pages.

In networks, the data is often noisy or lacking, with missing elements and imprecise relationships. Networks often have some type of structure: nodes are grouped into dense clusters and the node degree varies.

The challenge :

How to find latent coarse structure of a large, unlabeled network by using only the network topology?

Our approach:

- Latent component model based on a generative process.
- Depending on the hyperparameters, the latent properties are either latent blocks ("communities" in social networks) or more graded, hidden-variable-like structures.
- The model is fitted using collapsed Gibbs sampling.
- Sparse implementation allows feasible analysis of networks with millions of vertices and thousands of components.

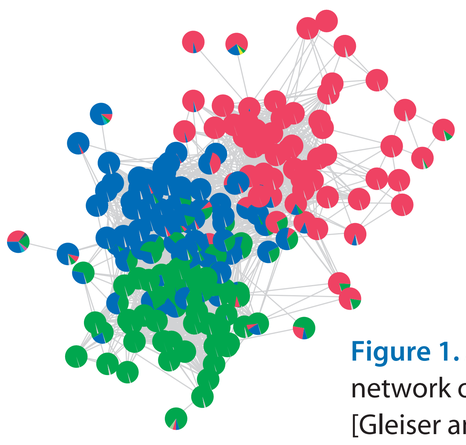

Figure 1. A collaboration network of jazz musicians [Gleiser and Danon, 2003] clustered with the latent component algorithm.

Model structure

In the model:

- Mixture components have a Dirichlet process prior.
- The model generates edges by drawing edge end points from distributions associated with each of the components.

The generative process underlying the model:

1. Draw θ from $DP(\alpha)$
2. For each component c in ∞ components
   (a) Draw m_c from $Dir(\beta)$
3. For each of E edges:
   (a) Draw a latent component z from θ
   (b) Draw first end point v_i from m_z
   (c) Draw second end point v_j form m_z

In this generative model, component probabilities θ arise from the Dirichlet process $DP(\alpha)$. Initially, a multinomial over N nodes is also set up for each component, by drawing successively from $Dir(\beta)$.

After m and θ have been set up, new edges can be generated by first picking a latent component z from θ and then selecting the edge endpoints v_i and v_j with probabilities m_c. By repeating this process for E times a network with E edges is obtained.

In Figure 2 the generative process is illustrated as a plate model.

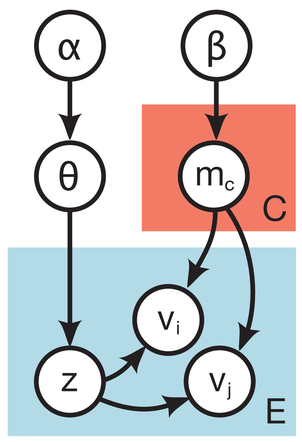

Figure 2. Plate representation of the generative model.

Results

An example of components found with the model is shown in Figure 1. The data set is jazz musicians from different cities in the US. The algorithm divided musician correcly into two cities. One of the cities further subdivided into two groups.

Figure 3 shows the clustering of a large friendship network of 147,610 Last.fm users claiming to be from the US, crawled from the Last.fm web site.

- For each user: demographics (age, country, sex) and music taste (artists).
- In addition, tags for over 188,565 artists were crawled.
- Clusters found on the basis of topology differ in terms of music tastes of the users.

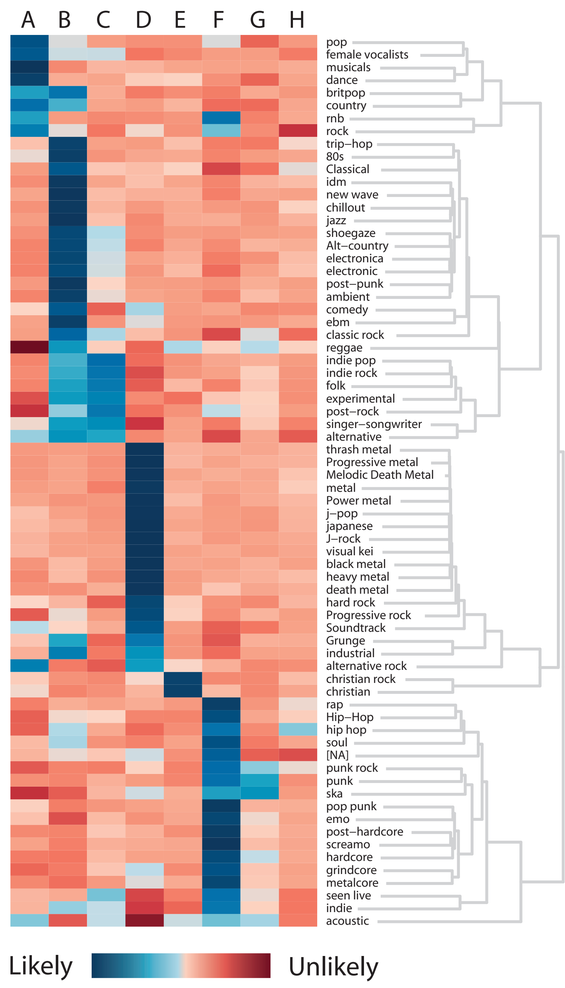

Figure 3. The Last.fm users from United States clustered into eight groups with their tags. The color of a cell depicts, how likely it is for a user who listens to a certain music tag (row) to belong to the particular cluster (column). Blue means that more than expected users belong to a cluster, while unpopular clusters among listeners of a specific tag are colored red. α=0.3 and β=0.3.

Component inference

The joint probability distribution of the model:

$$
\begin{aligned}
p_{DP}(L, Z, m|\alpha, \beta) &= \frac{\prod_{iz} m_{zi}^{\kappa_{zi}+\beta-1}}{D(E,\beta)^C} \times p(Z|n) \times p(n|\alpha) \\
&= \frac{\prod_{iz} m_{zi}^{k_{zi}+\beta-1}}{D(E,\beta)^C} \times \frac{\prod_z (n_z!)}{(2E)!} \times \frac{(2E)!\alpha^C}{C!\alpha^{[2E]}\prod_z n_z} \\
&= \frac{\prod_{iz} m_{zi}^{k_{zi}+\beta-1}}{D(E,\beta)^C} \times \frac{\alpha^C}{C!} \frac{\Gamma(\alpha)\prod_z \Gamma(n_z)}{\Gamma(\alpha+2E)} ,
\end{aligned}
$$

where $\alpha^{[2E]}$ is the Pochhammer symbol $x^{[n]} = \frac{\Gamma(x+n)}{\Gamma(x)}$

Z is the set of observed components for each edge, L is the set of edges, n is a vector containing counts for edges for each component, k_z is the vector of the number of edges in component z. Finally, $D(X,\xi) = \Gamma(\xi)^X/\Gamma(X\xi)$ is a normalizing term for the Dirichlet distribution.

The sampling of the edge components:

- Implemented with a collapsed Gibbs sampler.
- Unknown parameters m and θ marginalized away.
- Latent classes of the edges z are sampled one at a time.

The conditional probability for an edge cluster z_0 conditioned on all the other edges:

$$
\begin{aligned}
&p(z_0|i_0, j_0) \propto \\
&\frac{(k'_{z_0 i_0} + \beta)}{(2n'_{z_0} + 1 + N\beta)} \frac{(k'_{z_0 j_0} + \beta)}{(2n'_{z_0} + N\beta)} \frac{C(n'_{z_0}, \alpha)}{(E' + \alpha)} , \\
&C(n'_{z_0}, \alpha) = n'_{z_0} \text{ if } n'_{z_0} \neq 0 \text{ and } C(0, \alpha) = \alpha .
\end{aligned}
$$

k', n', and E' denote counts as if the link connecting nodes i_0 and j_0 would not exist at all.

Optimized implementation:

- Uses sparse data structures (hash tables, binary-tree representation of probabilities).
- When E and N are in the same order of magnitude, the average running time is essentially $\mathcal{O}(IE \log C)$. That is, excluding the number of iterations I, the running time scales linearly in the number of edges and logarithmically in the number of components.
- The memory consumption scales as $\mathcal{O}(L + C)$.

Conclusions

- Method is computationally efficient, however, sub-optimal hierarchical clustering methods are even faster.
- Provides information on the confidence of the clustering results.

Bibliography

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

J. Sinkkonen, J. Aukia, and S. Kaski. Inferring vertex properties from topology in large networks. In MLG'07 Working Notes. Universita degli Studi di Firenze, Florence, Italy. Extended Abstract.

Contacts:
janne.sinkkonen@xtract.com
janne.aukia@xtract.com
http://www.cis.hut.fi/projects/mi/papers/mlg07-13-sinkkonen-final.pdf