Inferring vertex properties from topology in large networks

*Janne Sinkkonen**, *Janne Aukia**, *and Samuel Kaski*†
*Xtract Ltd, Helsinki, Finland
†Laboratory of Computer and Information Science, Helsinki University of Technology, Finland

Abstract: Network topology not only tells about tightly-connected "communities," but also gives cues on more subtle properties of the vertices. We introduce a simple probabilistic latent-variable model which finds either latent blocks or more graded structures, depending on hyperparameters. With collapsed Gibbs sampling it can be estimated for networks of 10^6 vertices or more, and the number of latent components adapts to data through a Dirichlet process prior. Applied to the social network of a music recommendation site (Last.fm), reasonable combinations of musical genres appear from the network topology, as revealed by subsequent matching of the latent structure with listening habits of the participants. The advantages of the generative nature of the model are explicit handling of uncertainty in the sparse data, and easy interpretability, extensibility, and adaptation to applications with incomplete data.

Introduction

Data collections representable as networks, or sets of binary relations between vertices, appear now frequently in many fields. In this context, inferring properties of the vertices from the relations is a common data mining problem. We introduce an algorithm for the explorative case, where no *a priori* defined characteristics for the vertices are available and the task is to infer latent variables of the data. Rather than being predictive of a pre-defined target, the results are nominal or continuous-valued latent properties of the vertices that in a certain sense explain the observed connectivity well.

As a demonstration, musical tastes of people are derived from the friendship network of the online music servive Last.fm (`www.last.fm`), with over 10^5 vertices and edges. Other application areas besides social networks are found in biology (cellular networks) and hypertexts.

The algorithm introduced here is a simple generative probabilistic latent mixture model, fitted with (collapsed) Gibbs sampling [5]. Due to the Dirichlet process prior the complexity of the model is controlled by a parameter, but the number of latent components is not fixed. Careful implementation allows the analysis of millions of vertices with models including thousands of latent components. The generative nature of the model makes its description relatively explicit, including how uncertainty of data is taken into account. Extensions for richer relational information, missing data, and known properties of vertices are straightforward.

Methods

The generative process out of which the network is supposed to arise is the following; it is parameterized by (α, β)—see further below for discussion of these parameters, and Fig. 1 for a diagram: (1.1) Initialize by generating a multinomial distribution θ_z over an infinite number of *latent components* z from the Dirichlet process $\mathrm{DP}(Z|\alpha)$; (1.2) to each z, assign a multinomial distribution

over the M vertices i by sampling the multinomial parameters m_{zi} from the Dirichlet distribution Dir(β)—the prior is constant over the vertices. (To clarify, we have $\sum_i m_{zi} = 1$ for each z, and $\sum_z \theta_z = 1$.); (2) Then, repeat for each edge l: (2.1) draw a latent component z from the multinomial θ_z; (2.2) generate two vertices, i and j, independently of each other, with probabilities m_z; set up a non-directed edge between i and j.

Note that within components edges are generated independently of each other and "randomly"; the non-random structure of the network emerges from the tendency of components to prefer certain vertices (that is, m_z). In contrast to many other network models (e.g., [9] and stochastic block models [1, 4, 6, 7, 8]), latent variables operate on the edge level, not on the vertex level. There is no explicit hierarchy level for vertices, but because vertices typically have several edges, they are implicitly treated as mixtures over the latent components. Finally, the model is parameterized to generate self-links and multi-edges although they are not present in typical data sets. This allows sparse implementations that would be impossible for an equivalent Bernoulli model.

Although the number of potentially generated components is infinite, the Dirichlet process gives a very uneven distribution over them. Therefore, with a suitably small value of α, we observe much fewer components than the number of edges is, and the model is useful. On the other hand, β describes the unevenness of the degree distribution of the vertices *within components*: a high β tends to give components spanning over all vertices, while a small β prefers mutually exclusive, community-like components.

In the collapsed Gibbs sampling algorithm, the unknown model parameters m_{zi} and θ_z are marginalized away, and only the latent classes of the edges, z_l, are sampled one at a time. Given component assignment for all except the one edge to be sampled, denote edge counts per component by n_z, and component-wise vertex degrees by k_{zi}, and the endpoints of the left-out edge by (i, j). Then the component probabilities of the left-out edge are

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

with $\mathrm{C}(n_z, \alpha) \equiv n_z$ if $n_z \neq 0$ and $\mathrm{C}(0, \alpha) = \alpha$. In the latter case a new component is generated. This sampling step is simply repeated iteratively for all links, until convergence to the posterior distribution. Initialization of the sampler is most natural by starting from an empty urn, and populating it with edges on the first sampling round, again according to (1) but counting only the edges so far generated.

The sampling algorithm is easiest to derive by starting from a simpler model having a Dirichlet prior for z, then noting the structure of the collapsed sampling algorithm as two nested Polya urns (like in [5]), and replacing the urn for the components by the Blackwell-MacQueen urn [2] which corresponds to the Dirichlet Process prior of the present model.

Given the simplicity of the sampling scheme, it is easy to make it highly efficient. For example, the degree of a vertex poses an upper limit for its component heterogeneity, so that in most real-life networks only few of the counts k_{zi} are simultaneously non-zero, allowing sparse implementation and a high number of components. Edges can be sampled in parallel by locking vertices in the count

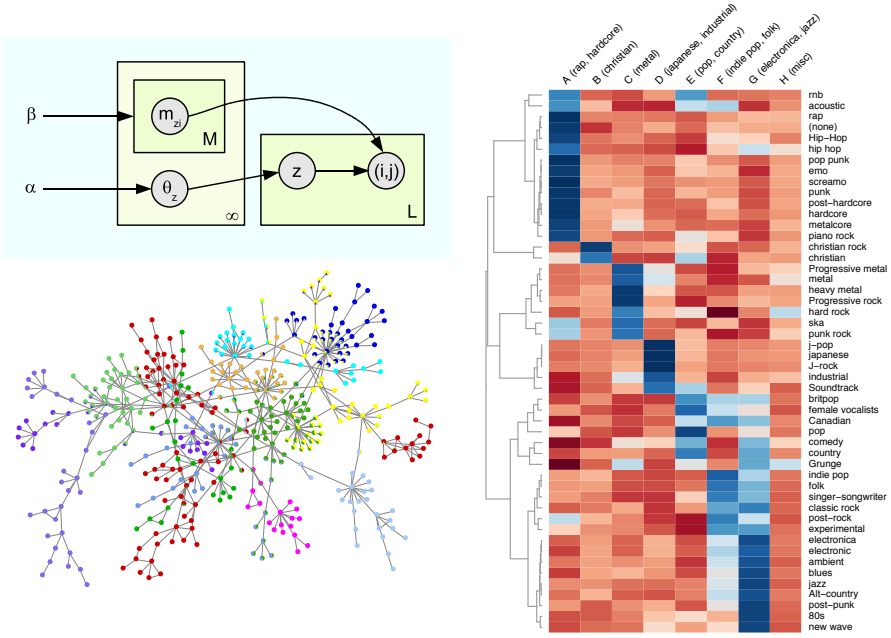

Figure 1: *Left, up:* The model. *Left, down:* Clusters, depicted by colors, found from a protein dataset similar to that in Airoldi et al. [1]. Note uncertainty in the middle. (Coordinates are from a directed-force layout with *Prefuse*.) *Right:* Last.fm user tastes derived from friendship network. Latent components were afterwards correlated with user's listening habits; here songs are aggregated by tags given to them. Users have mixed tastes, while tags are intuitively grouped into components. (Visualization with `heatplot` of *R*.)

table k_{zi}. For M vertices, L edges, C components, and I iteration rounds, memory consumption scales as $O(MC + L + C)$, but with hash tables this can be reduced to $O(Md + L + C)$ where d is the average degree. Because $d = L/M$, memory consumption scales as $O(L + C)$. After optimizing the sampling with trees, running time can be lowered down to $O(I\,L\,d\,\log C)$. That is, running time scales linearly in the number of edges and logarithmically in the number of components—excluding the required number of iterations I.

We also experimented with an EM algorithm for the finite-component version of the model. Althought it was faster than the MCMC method presented above, the results were clearly worse. Apparently, the posterior has many maxima at the borders of the parameter space, and the EM quickly converges into one of those.

Examples

A network of 455 proteins with 558 interactions is modelled with parameters (α=10^{-5}, β=10^{-4}) that prefer formation of discrete clusters. After 33,000 iteration rounds, 20 samples were taken at the intervals of 50 rounds, which resulted in 13 stable components, shown in Fig. 1, left. Clearly, the algorithm is capable of finding tightly connected components from the network.

The component structure reflecting musical tastes in Fig. 1, right, resulted from a run with 147,610 (anonymous) Last.fm users claiming to be from US, and their 352,987 self-announced mutual friendships. The parameters of the model (α=0.2, β=0.2) were chosen to prefer a small number of diffuse components. As a result (10,000 iterations in just under four hours), each user gets a probabilistic profile over eight latent components that seem to describe musical tastes. The tastes emerge from the friendships, which makes the approach usable as a customer relationship management and personalization tool.

Even larger models are feasible. Running a model of a social network with 670,000 users, about 3.5 million links and over 10,000 components into apparent convergence took 10,000 iterations, or 8 days of CPU on a single-processor Opteron.

Discussion

A simple generative algorithm seems to be able to capture both clusters and more diffuse latent properties of vertices, from network topology alone. Estimation with MCMC is surprising useful, replicating experiences with text models, where marginalized Gibbs is only 4–8 times slower and produces better results than the variational approach [3]. Obvious targets for futher work are proper empirical comparisons with other models and extension for richer datasets.

Acknowledgements: We thank Last.fm for making the social network and music data available, and Antti Ajanki (TKK) for preparing the protein data set. SK belongs to Helsinki Institute of Information Technology and Adaptive Informatics Research Centre, and is partly supported by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

[1] Airoldi EM, Blei DM, Fienberg SE, Xing EP (2006) Mixed-membership stochastic block models for relational data with application to protein-protein interaction. In *Proceedings of the International Biometrics Society Annual Meeting.*

[2] Blackwell D, MacQueen JB (1973) Ferguson distributions via polya urn schemes. *Annals of Statistics* 1, 353–355.

[3] Buntine W (2005) Dirichlet Processes for luddites: applications to discrete component analysis. *NIPS'05, workshop on non-parametric bayesian methods.*

[4] Daudin J-J, Picard F, Robin S (2007) A mixture model for random graphs. *Statistics for systems biology group, research report no. 4*, Jouy-en-Josas, France.

[5] Griffiths TL, Steyvers M (2004) Finding scientific topics. *PNAS* 101 suppl. 1, 5228–5235.

[6] Handcock MS and Raftery AE (2007) Model-based clustering for social networks. *J. R. Statist. Soc. A* 170, 1–22.

[7] Hoff PD, Raftery AE, Handcock MS (2002) Latent-space approaches to social-network analysis. *J. Am. Statist. Ass.* 97, 33–65.

[8] Kemp C, Griffiths TL, Tenenbaum JB (2004) Discovering latent classes in relational data. *MIT AI Memo* 2004-019.

[9] Newman MEJ, Leicht EA (2006) Mixture models and exploratory data analysis in networks. *arXiv:physics/0611158v2* (preprint).