An efficient block model for clustering sparse graphs*

Ádám Gyenge[†] [‡]
Computer and Automation Research Institute
Hungarian Academy of Sciences
adamgyenge@ilab.sztaki.hu

Janne Sinkkonen[‡]
Xtract Ltd.
Helsinki, Finland
janne.sinkkonen@gmail.com

András A. Benczúr
Computer and Automation Research Institute
Hungarian Academy of Sciences
benczur@ilab.sztaki.hu

ABSTRACT

Models for large, sparse graphs are found in many applications and are an active topic in machine learning research. We develop a new generative model that combines rich block structure and simple, efficient estimation by collapsed Gibbs sampling. Novel in our method is that we may learn the strength of assortative and disassortative mixing schemes of communities. Most earlier approaches, both based on low-dimensional projections and Latent Dirichlet Allocation implicitely rely on one of the two assumptions: some algorithms define similarity based solely on connectedness while others solely on the similarity of the neighborhood, leading to undesired results for example in near-bipartite subgraphs. In our experiments we cluster both small and large graphs, involving real and generated graphs that are known to be hard to partition. Our method outperforms earlier Latent Dirichlet Allocation based models as well as spectral heuristics.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database Applications—*data mining*

General Terms

Algorithms, Experimentation

Keywords

Block model, collapsed Gibbs, hierarchical Bayes, latent variables, statistical network analysis

*This work was supported by the EU FP7 project LiWA – Living Web Archives and by grants OTKA NK 72845 and NKFP-07-A2 *TEXTREND*.

[†]A part of the work of the author was done at the Budapest University of Technology and Economics, Hungary.

[‡]A part of the work of the author was done at the Department of Information and Computer Science, Aalto University School of Science and Technology, Helsinki.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MLG'10 July 24–25, 2010 Washington, DC, USA
Copyright 2010 ACM 978-1-4503-0214-2/10/07 ...$10.00.

1. INTRODUCTION

In recent years, the structure of large networks has become a popular research topic both in physics, statistics and machine learning and the search for statistically justified models that produce practical results in large networks of millions nodes or more remains open. Models are in particular applied for community analysis and network clustering [16].

Since the beginning, Latent Dirichlet Allocation (LDA) promised to be a candidate for network analysis [8]. LDA [5] is a popular latent class model, originally developed for document collections in the "bag of words" representation. Later it was applied in a wide variety of areas including image processing, web classification [4], and network clustering [21]. The latter is called the SSN-LDA, which uses the LDA on a document-document linking matrix. It assumes that each edge is generated from a latent topic with a distribution specific to the sending vertex. To each topic a distribution over the receiving vertices is associated. These topics form the latent "community" structure of the network, and links arise from parameters that are convex (non-negative) combinations of these communities. The parameters of LDA can be estimated with either a sampling or with variational techniques [2]. The collapsed Gibbs sampler is among the simplest and most popular estimation methods.

One of the biggest challenges is to develop methods that can be used also on large-scale networks that appear in the real data mining problems. In this paper we introduce a new model, the Interaction Dirichlet Block Model (IDBM) that combines the Poisson-like link model of LDA with a simple, non-mixing block structure. We apply collapsed Gibbs sampling for inference that allows sparse representation since it only processes existing edges. This makes estimation on large networks feasible and simple. Two sampling schemes are introduced, and both are implemented, optimized and tested. The block model is compared to other approaches on small test networks, on a medium size artificial network and on a large-scale data set.

An important concept of network clustering is the assortative mixing scheme or *homophily*: nodes are expected to have links with similar ones [15]. An opposite, bipartite-like concept is disassortative mixing, also called as *heterophily* or *homophoby*, in which the vertices form groups with the others that have different attributes. Prior to our work, graph partitioning methods typically used implicite assumptions on the mixing scheme and were unable to adapt to a particular class of graphs. For example, LDA is able to represent bipartite or disassortative elements with its latent structure since each node has a sender and a receiver role; these roles

are however not explicitely connected. As another example, the spectral method [7] approximates the adjacency matrix, in which nodes are represented by the vector of their neighbors and similarity corresponds to neighborhood overlap, involving an implicite, strong disassortative component.

The key element in our model is that we allow the nodes to play different roles in different interactions. Moreover, our model generated links not only between the members of the same community; the nodes may interact with nodes from other communities. We use our method via Gibbs inference and Bayesian estimation to obtain graph clustering. Our method outperforms SSN-LDA, spectral clustering, and two other models [1, 20] described next, both over small networks from Mark Newman's collection as described in Section 4.3, the WEBSPAM-UK2007 host graph [4], and generated graphs from a class believed to be hard to cluster [9].

1.1 Related results

A closely related model, the Interaction Component Model for Communities (ICMc), also represents the edge probabilities as convex combinations of communities, but the sending and the receiving probabilities are symmetrically sampled from the same multinomial distribution [20]. Here multinomials describe memberships probabilities of the vertices for a given component. Another multinomial represents the probability of a component. Hyperparameters act as priors for the multinomials and control the properties of the clustering: one influences component size, while the other controls the amount of component overlap. ICMc is capable of producing assortative grouping only.

The Mixed Membership Stochastic Block model (MMSB) [1] is a slightly different hierarchical generative model that allows more inter-group connections. It was originally proposed to explain the protein–protein interaction graph of cells. The structure of MMSB captures the different roles a protein can play in different biological processes. It is assumed that the proteins participate in many biological processes (latent components), and they can belong to different functional classes: in each reaction one protein present one function. These classes are called *blocks*, but they can be thought as latent components. It is important to note that this concept is different from both the assortative and disassortative mixing schemes, since it allows a general cohesion between the nodes based on all types of relations. The stochastic block model assigns two latent classes to each relationship and models the probability of reacting with each other for every pair of functional classes. In contrast to LDA and ICMc, not only links within the blocks but also links between the blocks are modeled. MMSB is parametrized with a Bernoulli mixing matrix between blocks. The structure of this model is very rich, but computational inference becomes challenging, and representation of sparsity requires additional tricks.

2. A BLOCK MODEL FOR SPARSE GRAPHS

In this section a new latent component mixture model is proposed for networks. Then, collapsed Gibbs sampling estimation algorithm with two different sampling schemes are developed for this new model to effectively infer the model parameters. The finite mixture model presented here merges the ideas of ICMc and MMSB.

The three generative models described in the previous section share many properties and also differ in many aspects. Efficient estimation algorithms allow to use LDA and ICMc on graphs having millions of nodes, but the formation of links in real networks is usually not restricted to either of the mixing schemes, since usually several processes lie behind the structure. MMSB introduces the concept of roles. Each of the nodes can play several roles in different context, and this is a good explanation for the mentioned richness of link establishment, the block structure. However, estimation of MMSB on very large graphs is difficult since it has too many parameters and the parametrization is not straightforward.

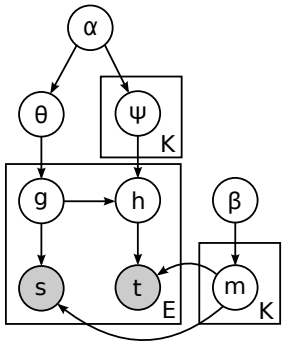

Figure 1: The graphical model representing the IDBM

While community models are suitable for many social networks and are easily parametrized, they assume tightly integrated subnetworks. Our Interaction Dirichlet Block Model (IDBM) is an extension of the ICMc which allows the nodes to play different roles in different reactions, similarly to as in the MMSB. In the concept of ICMc the links are established only between the members of the same communities, but the nodes can belong to more than one community. In the block models, the links between the communities (or blocks) are the organic part of the structure.

Similarly to the ICMc, a vector is associated to the latent components that describes how strongly the nodes belong to that component, or equivalently how well they can play the role of that block. The assumption behind IDBM is that links are generated first according to a latent distribution; then the link connects to a given node if it is able to play the corresponding role. This yields an explanation of linkage more powerful than that of community models.

2.1 Generative process

The Bayesian network of IDBM is shown in Figure 1. The generative process with K topics, E edges and V vertices is the following:

1. Draw $\vec{\theta} \sim Dir_K(\alpha)$, the probability of link generation
2. For each topic $k \in [1, K]$:
   (a) Draw $\vec{\psi}_k \sim Dir_K(\frac{\alpha}{K})$, the probability of receiving a link from another topic
   (b) Draw $\vec{m}_k \sim Dir_V(\beta)$, the topic membership probability
3. For each edge $e \in [1, E]$:

(a) Draw $g_e \sim Mult(\vec{\theta})$, the tail topic of link e

(b) Draw $h_e \sim Mult(\vec{\psi}_{g_e})$, the head topic of link e

(c) Draw $s_e \sim Mult(\vec{m}_{g_e})$, the tail node of link e

(d) Draw $t_e \sim Mult(\vec{m}_{h_e})$, the head node of link e

Role modeling is achieved by assigning two topics to each of the edges. The head and tail topics are the roles of the head and tail nodes, respectively. For a link, the head and tail nodes are sampled by taking the possible node roles into account. Similar to the MMSB, our IDBM models directed graphs but can be easily modified for undirected ones as well.

Parameters $\vec{\theta}$ and the matrix $\Psi = \{\vec{\psi}_k\}_{k=1}^K$ are key in the block model as they define the probability for each block to generate links, and the probability for all the topics to receive links. For instance, the probability for a new link to be generated by topic k is θ_k, and if this happens, then the probability for topic l to receive it is ψ_{kl}. If the rows of the matrix Ψ are multiplied by the corresponding coordinates of $\vec{\theta}$, then we get the cross linkage probability of the topics, a matrix equivalent to η in the MMSB.

3. PARAMETER ESTIMATION

Although our Interaction Dirichlet Block Model is relatively simple, similarly to LDA its exact inference is generally intractable. Since the dimension of the model is quite high, we propose the use of *collapsed Gibbs sampling*, a special case of Gibbs sampling where the parameters (in this case $\vec{\theta}$, Ψ and M) are marginalized out.

The two hyperparameters α and β contain the a priori knowledge about the latent components and the parameters. It is assumed that both of them are the same for all components and nodes, thus the model is symmetric. Nevertheless, it is possible to derive similar sampling formulas for asymmetric priors. In particular, the hyperparameter α controls the size of the components and also the correlation between them. For larger α, the sizes of the components are closer and inter-connectivity is strong. On the other hand, β represents the a priori component distributions of the nodes. The effect of a large β value is that a node is expected to belong to many components. In practice the variables $\vec{s}$ and $\vec{t}$ are observed, but for estimating the parameters $\vec{\theta}$, Ψ and M, the joint conditional distribution of the observed and latent variables is needed given the hyperparameters α and β.

3.1 Joint distribution

According to the graphical model, the joint distribution of the observed and latent variables is factoring to the following terms

$$p(\vec{g}, \vec{h}, \vec{s}, \vec{t}|M, \Psi, \vec{\theta}) = p(\vec{s}|\vec{g}, M)p(\vec{t}|\vec{h}, M)p(\vec{h}|\vec{g}, \Psi)p(\vec{g}|\vec{\theta}) \ , \tag{1}$$

where $M = \{\vec{m}_k\}_{k=1}^K$ and $\Psi = \{\vec{\psi}_k\}_{k=1}^K$.

The conditional distributions of the observed variables are

$$p(\vec{s}|\vec{g}, M) = \prod_{e=1}^{E} m_{g_e,s_e} = \prod_{k=1}^{K}\prod_{i=1}^{V} m_{ki}^{n_{ki}} \ , \tag{2}$$

$$p(\vec{t}|\vec{h}, M) = \prod_{e=1}^{E} m_{h_e,t_e} = \prod_{k=1}^{K}\prod_{i=1}^{V} m_{ki}^{p_{ki}} \ , \tag{3}$$

where n_{ki} refers to the number of times that node i has been observed as a source node for an edge with generator topic k, and p_{ki} refers to the number of times that node i has been observed as an end node for an edge with receiver topic k. Using these results the distribution of $\vec{s}$ and $\vec{t}$ given $\vec{g}$, $\vec{h}$, β can be calculated by marginalization

$$p(\vec{s}, \vec{t}|\vec{g}, \vec{h}, \beta) = \int p(\vec{s}, \vec{t}|\vec{g}, \vec{h}, M)p(M|\beta)\mathrm{d}M =$$

$$= \prod_{k=1}^{K} \frac{B(\vec{n}_k + \vec{p}_k + \beta)}{B(\beta)} \ , \tag{4}$$

where B is the multinomial Beta function.

The distribution of $\vec{g}$ given $\vec{\theta}$ can be written as

$$p(\vec{g}|\vec{\theta}) = \prod_{e=1}^{E} \theta_{g_e} = \prod_{k=1}^{K} \theta_k^{n_k} \ , \tag{5}$$

where $\vec{n}_k = \sum_{i=1}^{V} n_{ki}$ is the total number of times that the topic k has been observed as link generator, and similarly $\vec{p}_k = \sum_{i=1}^{V} p_{ki}$ is the total number of times that the topic k has been observed as link absorber. With this result, it is possible to derive the distribution of $\vec{g}$ given α by integrating out $\vec{\theta}$

$$p(\vec{g}|\alpha) = \int p(\vec{g}|\vec{\theta})p(\vec{\theta}|\vec{\alpha})\mathrm{d}\vec{\theta} = \frac{B(\vec{n} + \alpha)}{B(\alpha)} \ . \tag{6}$$

The third step is to get rid of Ψ. The distribution of $\vec{h}$ given $\vec{g}$ and Ψ is

$$p(\vec{h}|\vec{g}, \Psi) = \prod_{e=1}^{E} \psi_{g_e,h_e} = \prod_{k=1}^{K}\prod_{l=1}^{K} \psi_{kl}^{r_{kl}} \ , \tag{7}$$

where r_{kl} refers to the number of times that topic l was observed as end topic when start topic was k. Similarly to the first two steps, the distribution of $\vec{h}$ given $\vec{g}$ and α is achieved by marginalization

$$p(\vec{h}|\vec{g}, \alpha) = \int p(\vec{h}|\vec{g}, \Psi)p(\Psi|\alpha)\mathrm{d}\Psi = \prod_{k=1}^{K} \frac{B(\vec{r}_k + \frac{\alpha}{K})}{B(\frac{\alpha}{K})} \ . \tag{8}$$

Putting all together the joint distribution of the observable and latent variables is

$$p(\vec{s}, \vec{t}, \vec{g}, \vec{h}|\alpha, \beta) = p(\vec{s}, \vec{t}|\vec{g}, \vec{h}, \beta)p(\vec{g}|\alpha)p(\vec{h}|\vec{g}, \alpha) =$$

$$\frac{B(\vec{n} + \alpha)}{B(\alpha)} \prod_{k=1}^{K} \frac{B(\vec{n}_k + \vec{p}_k + \beta)}{B(\beta)} \frac{B(\vec{r}_k + \frac{\alpha}{K})}{B(\frac{\alpha}{K})} \ . \tag{9}$$

3.2 One-phase collapsed Gibbs sampling

An efficient collapsed Gibbs sampling procedure is needed to sample sequentially the two latent components for each edge from the conditional distribution of that edge given all the other links and component assignments in the network. In this case, the variables g and h are treated as one, two dimensional random variable for each edge. This latent variable is sampled from a two dimensional multinomial distribution.

Since $r_{g_e} = n_{g_e}$, the conditional probability of link e given the rest of the complete data is

$$p(s_e, t_e, g_e, h_e|\vec{s}_{\neg e}, \vec{t}_{\neg e}, \vec{g}_{\neg e}, \vec{h}_{\neg e}) =$$

One-Phase-GS	Two-Phase-GS
Initialization	Initialization
For $i \in [1, I]$	For $i \in [1, I]$
For each edge $e \in [1, E]$	For each edge $e \in [1, E]$
Sample (g_e, h_e)	Sample $g_e \sim p(g_e \vert \dots)$
$\sim p(g_e, h_e \vert \dots)$	Sample $h_e \sim p(h_e \vert \dots)$

Table 1: The one- and two-phase Gibbs sampling algorithms for IDBM

$$\frac{r_{g_e h_e} + \frac{\alpha}{K}}{E + K\alpha} \cdot \frac{(n_{g_e s_e} + p_{g_e s_e} + \beta)(n_{h_e t_e} + p_{h_e t_e} + \beta)}{(n_{g_e} + p_{g_e} + V\beta)([n_{h_e} + p_{h_e} + V\beta] - \delta_{g_e h_e})} . \tag{10}$$

3.3 Two-phase collapsed Gibbs sampling

In contrast to the one-phase Gibbs sampler developed in the previous section that samples the two latent variables g and h simultaneously from the joint conditional distribution, it is possible to sample the two latent variables g and h separately. In this case, for each edge g is drawn first from its one-dimensional conditional distribution, and then similarly h is sampled in the second phase of each iteration.

The conditional probability for the absorbing component of link e given the rest of the network is

$$p(t_e, h_e | \vec{s}_{\neg e}, \vec{t}_{\neg e}, \vec{g}_{\neg e}, \vec{h}_{\neg e})$$

$$= \int\int p(s_e, t_e, g_e, h_e | \vec{s}_{\neg e}, \vec{t}_{\neg e}, \vec{g}_{\neg e}, \vec{h}_{\neg e}) \mathrm{d}s \mathrm{d}g =$$

$$= \frac{n_{h_e t_e} + p_{h_e t_e} + \beta}{[n_{h_e} + p_{h_e} + V\beta] - \delta_{g_e h_e}} \cdot \frac{p_{h_e} + \alpha}{E + K\alpha} . \tag{11}$$

Using this it is possible to calculate the conditional probability of the generating component for each edge

$$p(s_e, g_e | \vec{s}_{\neg e}, \vec{t}, \vec{g}_{\neg e}, \vec{h}) = \frac{p(s_e, t_e, g_e, h_e | \vec{s}_{\neg e}, \vec{t}_{\neg e}, \vec{g}_{\neg e}, \vec{h}_{\neg e})}{p(t_e, h_e | \vec{s}_{\neg e}, \vec{t}_{\neg e}, \vec{g}_{\neg e}, \vec{h}_{\neg e})} =$$

$$= \frac{r_{g_e h_e} + \frac{\alpha}{K}}{p_{h_i} + \alpha} \cdot \frac{n_{g_e s_e} + p_{g_e s_e} + \beta}{n_{g_e} + p_{g_e} + V\beta} , \tag{12}$$

where h_e is the current latent component absorbing the edge. Similarly, the conditional probability of the absorbing component is

$$p(t_e, h_e | \vec{s}, \vec{t}_{\neg e}, \vec{g}, \vec{h}_{\neg e}) = \frac{r_{g_e h_e} + \frac{\alpha}{K}}{n_{g_e} + \alpha} \cdot \frac{n_{h_e t_e} + p_{h_e t_e} + \beta}{n_{h_e} + p_{h_e} + V\beta} , \tag{13}$$

where g_e is the current latent component generating the edge.

For each edge the sampling of the two latent variables requires $\mathcal{O}(K^2)$ steps in the case of the one-phase scheme, while in the two-phase method only $\mathcal{O}(K)$ steps are required. If the number of iterations is I, then the total running time of the first and second cases is $\mathcal{O}(IEK^2)$ and $\mathcal{O}(IEK)$, respectively. Thus, the two-phase method is comparable to the standard implementation of ICMc and LDA, and in the case of sparse graphs it outperforms the $\mathcal{O}(IV^2K)$ time MMSB. On the other hand, without any further optimization it is slower than the best implementations of ICMc and LDA [18], [3].

4. EXPERIMENTS AND RESULTS

Initial tests confirmed what theory suggests about the relative performance of the two samplers: the two-phase sampler was much more efficient in terms of both speed and memory consumption. Because it also converged well, the two-phase algorithm was chosen to be evaluated and be compared to other methods.

The experiments were carried out on three different types of graphs using four different models: IDBM with two-phase Gibbs sampler, ICMc, LDA and MMSB. We used our own implementations of IDBM and ICMc[1], and for the other models the R package 'lda' developed by Jonathan Chang [6]. It contains collapsed Gibbs samplers for LDA and MMSB, and is freely available. As it turns out from the tests, the sampler for LDA is fast enough, but the MMSB sampler is much slower than the others. It was able to run only on the smallest graphs with a few number of components in a reasonable time.

In addition to the experiments with all four models on a set of small graphs, the faster LDA, ICMc, and IDBM, together with a spectral clustering algorithm, were tested on a medium-size artificial graph, which has properties that makes it difficult to cluster with classic algorithms [10] (see section 4.4).

Last, LDA, ICMc, IDBM and spectral clustering were compared on a large web graph. In all of the experiments we took the most likely (MAP) division of the nodes given an average over the samples in the case of the sampling algorithms.

4.1 Clustering quality measures

The quality of a graph partitioning can be measured in several ways. These measures belong to two classes. In the first case we only use information encoded in the graph itself, while in the second case we use external ground truth, the natural clusters of the nodes.

Modularity, one of the most common graph cut quality functions, measures the ratio of intra-cluster edges minus the square of the ratio of the edges ending up in the cluster. The directed version defined in [12] is more appropriate in the cases of directed graphs while it applies unchanged to undirected networks as well. The modularity of a clustering of a directed network is defined as

$$Q = \sum_{\text{clusters } s} \left[\frac{|E(C_s, C_s)|}{|E|} - \frac{|E(C_s, \overline{C_s})|}{|E|} \cdot \frac{|E(\overline{C_s}, C_s)|}{|E|} \right] , \tag{14}$$

where E is the set of all edges and $E(X, Y)$ is the set of edges with tail in X and head in Y. Shiga et al. [19] pointed out that modularity is not balanced by the size of the clusters, which means that in a clustering with high modularity a group might become small when affected by outliers. They proposed a slightly modified version, called the *normalized modularity*, which is balanced by the cluster size and for directed networks it is defined as

$$Q' = \sum_{\text{clusters } s} \frac{N}{N_s} \left[\frac{|E(C_s, C_s)|}{|E|} - \frac{|E(C_s, \overline{C_s})|}{|E|} \cdot \frac{|E(\overline{C_s}, C_s)|}{|E|} \right] , \tag{15}$$

where N is the total number of nodes and N_s is the number of vertices in the cluster s. The larger the modularity or the normalized modularity is, the more edges are connecting

[1] `http://www.ilab.sztaki.hu/~adamgyenge/mcnet/`

the members of the same cluster and the less connect the members of the different clusters, and therefore the edges are more dense in the groups.

In the case of graph clustering, the estimated and the true clusters can be treated as random variables. A contingency table N can be formed, where N_{ij} is the number of elements that belong to cluster i in the estimation and to cluster j in the ground truth. The *variation of information* is defined in [13] as

$$d_{VI}(X,Y) = H(X)+H(Y)-2I(X,Y) = H(X|Y)+H(Y|X)\,, \tag{16}$$

where $H(X)$ and $H(Y)$ are the (absolute) entropy, $H(X|Y)$ and $H(Y|X)$ are the relative entropies of the estimated and true clustering, respectively, and $I(X,Y)$ is the mutual information between them. The variation of information satisfies several naturally arising axioms for (clustering) distances. In particular, it is a *metric* on the set of all partitions of a graph. Therefore, it is convenient to use the variation of information as the measure to compare the estimated and the true clustering instead of other entropy based measures, e.g. perplexity. While higher mutual information means better clustering, for the variation of information (or VI distance) the lower value shows proximity to the ground truth.

4.2 Initialization and convergence

In the initialization step for the sampler of the IDBM and ICMc, edges were randomly associated to the topics independently of each other. This is a conservative initialization method—more elaborate procedures are likely to lead to better convergence. Especially starting with a small proportion of the edges and incrementally inserting them to the simulation is likely to better avoid local minima.

Convergence of the chains was monitored by two kinds of measures: the two modularities (ordinary and normalized), and the incomplete data log-likelihood, defined as

$$\log p(\vec{s},\vec{t}|\vec{g},\vec{h},\beta) = K(\log\Gamma(V\beta) - V\log\Gamma(\beta))-$$

$$-\sum_{k=1}^{K}\left[\log\Gamma(n_k+p_k+V\beta) - \sum_{l=1}^{V}\log\Gamma(n_{kl}+p_{kl}+\beta)\right] \tag{17}$$

for the IDBM. This leave-one-out predictive likelihood is used mainly because it is easy to compute during the simulation by numerically approximating the $\log\Gamma$ function. In the experiments, it turned out that both of them can indicate the convergence of the chain. The modularities are applied to the most likely division of the nodes according to the current status of the chain. In practice, the chain reaches the stationary state after 100–200 iterations in the case of smaller graphs, and interestingly after 50–100 iterations for larger graphs.

4.3 Small networks

We applied the four Bayesian models on selected graphs of Mark Newman with ground truth available[2]. The Karate network originates from a study on the social relations in a karate club with 34 members that later split. Out of the 78 relations 10 are between the two clusters. The Football network represents the schedule of American football games in Division I of colleges in the 2000 season. The 115 vertices in the graph represent teams, and 613 edges between them represent regular-season games between the teams they connect. This network also has a known structure of subgroups. Teams are divided into 12 conferences containing around 8-12 teams each. Games are more frequent between the members of the same conference than between the members of different conferences. The graph has 219 inter-conference edges. The Jazz network lists 2432 collaborations of jazz bands between 1912 and 1940. The 187 nodes of the graph are jazz groups and two of them are connected if they have a musician in common. The partitions of the ground truth are based on the 7 locations where the bands had recorded (Chicago, New York, etc.). The graph has 1600 inter-cluster edges. The Polblogs network is connected to the 2004 U.S. Presidential Election, which was the first Presidential Election in the United States in which blogging played an important role. 19089 hyperlinks between 1222 political blogs were automatically extracted from a crawl of the front page of the blogs during the campaign. The nodes of this directed graph are categorized into the groups of the principally conservative and the principally liberal blogs (1688 inter-cluster edges).

The hyperparameters for highest normalized modularity or likelihood can be obtained by exhaustive search for all algorithms. Another approach is to define hyperpriors on the hyperparameters, and sample them as well. In [2] the two methods are compared and evaluated, and it is showed that there is no major difference between the results. Therefore, for simplicity we used the exhaustive search method. Preliminary tests had shown that reasonable values for the IDBM algorithm lie in the range of $\alpha = 0.001\ldots1000$ and $\beta = 0.001\ldots100$. Based on the original papers, these intervals are the same for LDA and ICMc and, by our measurements, for the MMSB $\alpha = 0.001\ldots100$ and $\beta = 1\ldots1000$ should be appropriate intervals to search through. In the cases when the ground truth is available, the number of latent components were set to the real number of clusters. It turns out that adding some extra components does not significantly influence the results. On the other hand, during the sampling some components may occasionally vanish, thus the number of clusters in the result may be lower than the number of latent components.

To compare our results with the ground truth, the VI distance of the true and the estimated clusters were calculated in each case. The results were slightly worse when the parameters were optimized for likelihood than for modularity. Table 2 shows the latter cases. It seems that the normalized modularity is a better graph-only measure for these networks generally, because the optimal results in this case have a lower VI distance to the real grouping. On the other hand, in the case of the Jazz network the best result is given by IDBM when optimized for modularity. Except for the smallest networks none of the results are perfect. This might be because of both measures are connected to the assortative mixing scheme. Nevertheless, without any additional information about the graph, the only way is to optimize with respect to the homophilic assumption, and especially for normalized modularity. It can be seen that on these networks the IDBM outperforms the other three methods, since it achieves the smallest VI distances. Still, the results produced by the ICMC usually have higher modularity.

It is important to mention that in our experiments, IDBM is particularly good at estimating the number of inter-cluster

[2] `http://www-personal.umich.edu/~mejn/netdata/`

	LDA		MMSB		ICMc		IDBM	
	Q	d_{VI}	Q	d_{VI}	Q	d_{VI}	Q	d_{VI}
Karate	0.371	0	0.332	0.847	0.371	0	0.371	0
Football	0.573	0.872	0.374	2.108	0.603	0.794	0.603	0.671
Jazz	0.428	3.531	0.329	4.354	0.434	3.583	0.433	**3.327**
Polblogs	0.430	0.570	-	-	0.430	0.570	0.431	0.576
	Q'	d_{VI}	Q'	d_{VI}	Q'	d_{VI}	Q'	d_{VI}
Karate	0.745	0	0.688	0.645	0.745	0	0.745	0
Football	6.440	0.636	4.243	2.108	6.951	0.566	6.922	**0.546**
Jazz	2.239	4.232	1.992	4.279	1.863	3.881	1.975	3.911
Polblogs	0.868	0.608	-	-	0.868	0.570	0.868	**0.549**

Table 2: The highest modularities and the corresponding VI distances of the results with each algorithm. On the top we optimize for modularity (Q) while on the bottom for normalized modularity (Q') and show the VI distance for the modularity based optimum.

edges. In the three larger cases they were around 200, 730 and 1290, while the ICMc and the LDA usually achieved by 20-40 edges less. The number of inter-cluster edges in the case of the MMSB were higher but this was at the expense of poor estimation of the true clusters compared to the other three methods.

4.4 Artificial graphs that are difficult to cluster

It has recently turned out that classical spectral clustering methods usually fail on real social networks [11]. This is mainly due to some special structural properties of such graphs like the presence of tightly knit communities and long tentacles around a big core. Kurucz et al. proposed a modification of spectral clustering with heuristical redistribution and vertex contraction steps [10]. We compared the LDA, ICMc and IDBM to this advanced spectral clustering algorithm involving several heuristics.

An artificial scale-free network of 9983 nodes and 28051 edges was created according to a power-law small-world model, defined as follows [9]. The starting point is the small world graph model with nodes placed over a 2D grid. Next, geographically dense regions over the grid are generated by assigning density to each node according to a power law distribution with exponent -1.33. Finally, nodes are connected with probability inversely proportional to their squared Euclidean distance. In this model for each vertex a number t by a power law distribution with exponent -1.33 is generated and t edges are added independent with probability.

Graphs generated with this model are hard to partition with the classical spectral methods. We compared the four Bayesian models to the advanced heuristic spectral clustering on the generated graph. The number of clusters (topics) were set to 4, and the hyperparameters were optimized with exhaustive search (SVD was performed in 4 dimension). The results are summarized in Table 3. It can be seen that LDA and ICMc produce higher modularities, but SVD and especially the IDBM are better in terms of normalized modularity. One reason for this can be that these algorithms can produce more varied-sized clusters which allows higher normalized modularity.

4.5 Clustering the web graph

On a Web subgraph, we managed to compared the four most efficient algorithms in order to investigate their performance on larger networks and the analyze the possibility of using them in Web applications. Generally, hypertext analysis may make use of document terms as well as linkage. We investigate link-only versions the models. Nonetheless, it is possible to extend all models including our IDBM by modeling the words similarly as proposed in e.g. [14]. The data set used was the largest connected component from the public test data of Web Spam Challenge 2007 [4]. This network consists of 111083 nodes and 1836338 edges. One node represents a host from the `.uk` domain and one directed edge points from one node to another if the first hosts a Web page having a hyperlink to a page hosted by the second node.

	Q	Q'
SVD + heuristics	0.551	2.516
LDA	**0.573**	2.339
MMSB	-	-
ICMc	**0.567**	2.272
IDBM	0.527	**2.582**

Table 3: Highest modularities on a difficult-to-cluster artificial scale-free network.

The Open Directory project[3] contains human edited categorization of more than 4 million Web page. From this database the 14 top level categories were used to label the nodes of the UK-host graph. If a site contained a page registered in DMOZ with some top category, then it was labeled with that category. In case of a conflict a random page of the site registered in DMOZ was chosen and its site was labeled with its top category. In this way category labels for 38415 hosts have been obtained. These labels were used to analyze the structure of the clusters, but since not all of the nodes were labeled, the computation of the full VI distance was not possible.

For such large-scale data sets it is not possible to perform a full exhaustive search in the three-dimensional parameter space of α, β and K. Based on initial results and the previous sections we concluded that the clustering of larger graphs is quite robust with respect to the hyperparameters, especially to α. Thus, the hyperparameters were selected from a smaller interval: $0.01 \dots 1$. The number of latent components has been set to $K = 2 \dots 20$. The modularities of the clustering produced by IDBM as functions of number of topics are plotted in Figure 2. It seems that neither

[3] `www.dmoz.org`

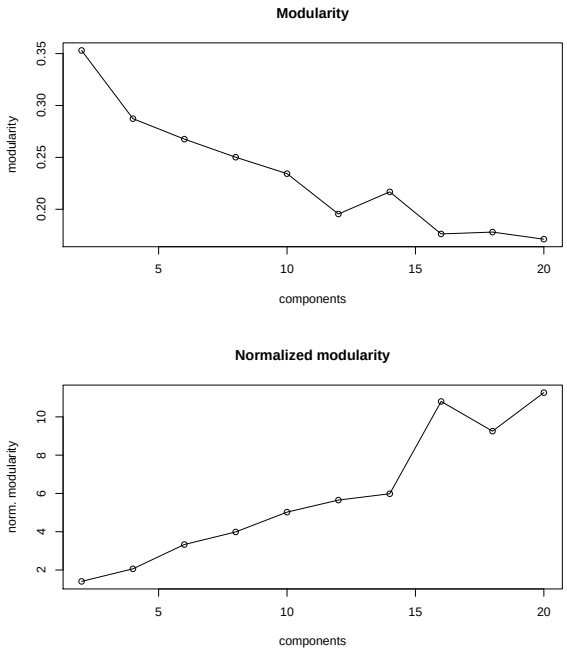

Figure 2: Modularity and normalized modularity of the results produced by IDBM on the UK-host graph.

of these measures indicate the optimal number of clusters, because they decrease or increase with the number of topics. As mentioned above, the number of top categories in the DMOZ directory is 14. We used also $K = 14$ topics in the forthcoming test to compare the labels and the clusters. The SVD was performed in 10 dimensions.

We created two-way contingency tables by considering hosts for which at least one DMOZ label is available. The tables have the labels as rows and the estimated groups as columns. Using these tables, the estimated clusters were compared to the manually associated labels in the terms of VI distance. The quality of the divisions produced by the different methods is shown in Table 4. Interestingly, in this network the best result is achieved with the generative models optimized for log-likelihood. On the other hand, the purely graph based measures differ significantly, while the VI distance of estimated clusters to the labels is quite similar in these cases. For example, although Q and especially Q' are the smallest in the case of the SVD, it achieves the one of the best results. As it seems, the lowest VI distance is produced by IDBM optimized for log-likelihood. This was obtained by setting $\alpha = 0.1$, $\beta = 0.66$, and by taking 10–150 samples with a sample lag of 5 iterations. The running time of the Gibbs sampling was around 10 minutes in this case, and the likelihood stabilized around $-2.4 \cdot 10^6$.

The heat map of the two-way contingency table produced by the IDBM is plotted in Figure 3. As it can be seen, the clustering is not able to separate the labels. We observe that some groups of components strongly correlate with certain labels. For example, Science, Reference and Health are basically concentrated on groups A, C, H, while Games, Shopping, Business and Computer mostly occur in the clusters L, K, I and M. Similarly, the groups E and N cover the

	Q	d_{VI}	Q'	d_{VI}	LL	d_{VI}
SVD	0.221	4.682	3.654	4.682	-	-
LDA	0.255	5.901	11.61	5.825	-1.2e7	4.268
MMSB	-	-	-	-	-	-
ICMc	0.381	5.961	15.22	5.805	-2.5e6	4.927
IDBM	0.241	6.103	7.349	5.972	-2.4e6	**4.144**

Table 4: The highest modularities, log-likelihoods and the corresponding VI distances of the results on the UK host graph with 14 latent topics (clusters).

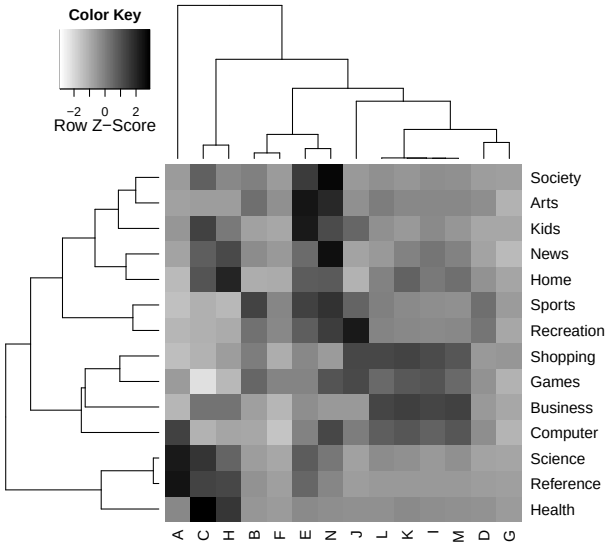

Figure 3: Heat map of the contingency table between the labels and the clusters produced by the IDBM on the UK host graph.

most of Society, Arts, Kids and News, while Recreation and component J are also highly correlated. The size of the predicted clusters ranges from 3000 to 10000. Although the VI distance of 4.1–5.0 is quite high on the absolute scale, it is relatively low concerning the number of nodes. One might argue that the results are still insufficient for the automatic categorization of web pages, but we might also note that the categories themselves are overlapping. Indeed, a large number of Web sites belong to multiple DMOZ categories. In any case, we believe that the result of the clustering can serve as a useful information for Web crawlers and Web directory editors.

5. DISCUSSION AND CONCLUSIONS

We have combined features of earlier probabilistic graph clustering approaches to a new, simple block model that is easily applicable to large, sparse networks, especially if estimated with collapsed Gibbs sampling. The new model was compared to community models and to another block model on a set of small test networks, as well as to a spectral clustering algorithm on a web host graph of the `.uk` domain and on an artificial data set. The results of the web clustering

were measured in terms of manual labeling. This complements some limited preliminary results of a similar model already presented in [17].

In order to obtain high quality clusters, we have to carefully initialize the sampler and set the hyperparameters. Better initial mixing may be achieved if the edges are added to the simulation gradually. Hyperparameters can be estimated as for LDA. Infinite component variants of the model would be possible by using a hierarchy of Dirichlet priors.

The difference between community and block models is relatively small. Both type of models assign latent categorical variables to the edges, and the memberships of the vertices in the clusters are computed based on the latent variables of the edges connecting to node. Community models assign only one latent variable per edge, while block models assign two latent variables to each edge. In the latter, the two variables are bound by an interaction matrix. This interaction matrix is a good parametrization when the graph is not expected to follow an assortative structure, that is when it has "bipartite properties", or communities have complex interactions that we want to estimate, or that are necessary to estimate to reveal the community structure. The four algorithms usually produce similar results on the small networks with a very good performance of IDBM in terms of the VI distance. It has also turned out that modularity is not the best cost function for clustering in many real-world graphs. In the case of large graphs and our algorithm, likelihood turned out to produce better parameter estimation.

Acknowledgments

The authors wish thank to Samuel Kaski, Juuso Parkkinen, Miklós Kurucz, István Bíró and Zoltán Gyöngyi for their useful comments and datasets.

6. REFERENCES

[1] E. Airoldi, D. M. Blei, E. P. Xing, and S. Fienberg. A latent mixed membership model for relational data. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 82–89. ACM, 2005.

[2] A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh. On Smoothing and Inference for Topic Models. In *UAI 2009: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.

[3] J. Aukia. Bayesian clustering of huge friendship networks. Master's thesis, Helsinki University of Technology, 2007.

[4] I. Bíró, J. Szabó, and A. A. Benczúr. Latent dirichlet allocation in web spam filtering. In *AIRWeb '08: Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, New York, NY, USA, 2008. ACM.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] J. Chang. *LDA: Collapsed Gibbs sampling methods for topic models*, 2010. R package version 1.2.

[7] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

[8] M. Girolami and A. Kaban. Sequential activity profiling: latent Dirichlet allocation of Markov chains. *Data Mining and Knowledge Discovery*, 10(3):175–196, 2005.

[9] M. Kurucz and A. A. Benczúr. Geographically organized small communities and the hardness of clustering social networks. *Annals of Information Systems - Data Mining for Social Network Analyis, to appear. Available at* `http://datamining.sztaki.hu/files/AoIS.pdf`, 2010.

[10] M. Kurucz, A. A. Benczúr, and A. Pereszlényi. Large-scale principal component analysis on LiveJournal friends network. *Proceedings of SNAKDD*, 2008.

[11] Kevin Lang. Fixing two weaknesses of the spectral method. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *NIPS '05: Advances in Neural Information Processing Systems 18*, pages 715–722, Cambridge, MA, 2006. MIT Press.

[12] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Physical Review Letters*, 100(11):118703, 2008.

[13] M. Meilă. Comparing clusterings: an axiomatic view. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 577–584, New York, NY, USA, 2005. ACM.

[14] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 542–550. ACM, 2008.

[15] M. E. J. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126, 2003.

[16] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

[17] J. Parkkinen, A. Gyenge, J. Sinkkonen, and S. Kaski. A block model suitable for sparse graphs. In *MLG 2009, The 7th International Workshop on Mining and Learning with Graphs*, 2009.

[18] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM, 2008.

[19] M. Shiga, I. Takigawa, and H. Mamitsuka. A spectral clustering approach to optimally combining numerical vectors with a modular network. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 647–656, New York, NY, USA, 2007. ACM.

[20] J. Sinkkonen, J. Aukia, and S. Kaski. Inferring vertex properties from topology in large networks. In *Working Notes of the 5th International Workshop on Mining and Learning with Graphs (MLG'07)*, 2007.

[21] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An LDA-based community structure discovery approach for large-scale social networks. In *Intelligence and Security Informatics (ISI) 2007*, pages 200–207. IEEE, 2007.