The Annals of Applied Statistics
2013, Vol. 7, No. 2, 1010–1039
DOI: 10.1214/12-AOAS617

MODEL-BASED CLUSTERING OF LARGE NETWORKS[1]

BY DUY Q. VU, DAVID R. HUNTER AND MICHAEL SCHWEINBERGER

University of Melbourne, Pennsylvania State University and Rice University

We describe a network clustering framework, based on finite mixture models, that can be applied to discrete-valued networks with hundreds of thousands of nodes and billions of edge variables. Relative to other recent model-based clustering work for networks, we introduce a more flexible modeling framework, improve the variational-approximation estimation algorithm, discuss and implement standard error estimation via a parametric bootstrap approach, and apply these methods to much larger data sets than those seen elsewhere in the literature. The more flexible framework is achieved through introducing novel parameterizations of the model, giving varying degrees of parsimony, using exponential family models whose structure may be exploited in various theoretical and algorithmic ways. The algorithms are based on variational generalized EM algorithms, where the E-steps are augmented by a minorization-maximization (MM) idea. The bootstrapped standard error estimates are based on an efficient Monte Carlo network simulation idea. Last, we demonstrate the usefulness of the model-based clustering framework by applying it to a discrete-valued network with more than 131,000 nodes and 17 billion edge variables.

1. Introduction. According to Fisher [(1922), page 311], "the object of statistical methods is the reduction of data." The reduction of data is imperative in the case of discrete-valued networks that may have hundreds of thousands of nodes and billions of edge variables. The collection of such large networks is becoming more and more common, thanks to electronic devices such as cameras and computers. Of special interest is the identification of influential subsets of nodes and high-density regions of the network with an eye to break down the large network into smaller, more manageable components. These smaller, more manageable components may be studied by more advanced statistical models, such as advanced exponential family models [e.g., Frank and Strauss (1986), Hunter and Handcock (2006), Snijders et al. (2006), Strauss and Ikeda (1990), Wasserman and Pattison (1996)].

Received May 2012; revised November 2012.

[1]Supported in part by the Office of Naval Research (ONR Grant N00014-08-1-1015) and the National Institutes of Health (NIH Grant 1 R01 GM083603). Experiments in this work were also supported in part through instrumentation funded by the National Science Foundation (NSF Grant OCI-0821527).

Key words and phrases. Social networks, stochastic block models, finite mixture models, EM algorithms, generalized EM algorithms, variational EM algorithms, MM algorithms.

An example is given by signed networks, such as trust networks, which arise in World Wide Web applications. Users of internet-based exchange networks are invited to classify other users as either -1 (untrustworthy) or $+1$ (trustworthy). Trust networks can be used to protect users and enhance collaboration among users [Kunegis, Lommatzsch and Bauckhage (2009), Massa and Avesani (2007)]. A second example is the spread of infectious disease through populations by way of contacts among individuals [Britton and O'Neill (2002), Groendyke, Welch and Hunter (2011)]. In such applications, it may be of interest to identify potential super-spreaders—that is, individuals who are in contact with many other individuals and who could therefore spread the disease to many others—and dense regions of the network through which disease could spread rapidly.

The current article advances the model-based clustering of large networks in at least four ways. First, we introduce a simple and flexible statistical framework for parameterizing models based on statistical exponential families [e.g., Barndorff-Nielsen (1978)] that advances existing model-based clustering techniques. Model-based clustering of networks was pioneered by Snijders and Nowicki (1997). The simple, unconstrained parameterizations employed by Snijders and Nowicki (1997) and others [e.g., Airoldi et al. (2008), Daudin, Picard and Robin (2008), Mariadassou, Robin and Vacher (2010), Nowicki and Snijders (2001), Zanghi et al. (2010)] make sense when networks are small, undirected and binary, and when there are no covariates. In general, though, such parameterizations may be unappealing from both a scientific point of view and a statistical point of view, as they may result in nonparsimonious models with hundreds or thousands of parameters. An important advantage of the statistical framework we introduce here is that it gives researchers a choice: they can choose interesting features of the data, specify a model capturing those features, and cluster nodes based on the specified model. The resulting models are therefore both parsimonious and scientifically interesting.

Second, we introduce approximate maximum likelihood estimates of parameters based on novel variational generalized EM (GEM) algorithms, which take advantage of minorization-maximization (MM) algorithms [Hunter and Lange (2004)] and have computational advantages. For unconstrained models, tests suggest that the variational GEM algorithms we propose can converge quicker and better avoid local maxima than alternative algorithms; see Sections 6 and 7. In the presence of parameter constraints, we facilitate computations by exploiting the properties of exponential families [e.g., Barndorff-Nielsen (1978)]. In addition, we sketch how the variational GEM algorithm can be extended to obtain approximate Bayesian estimates.

Third, we introduce bootstrap standard errors to quantify the uncertainty about the approximate maximum likelihood estimates of the parameters, whereas other work has ignored the uncertainty about the approximate maximum likelihood estimates. To facilitate these bootstrap procedures, we introduce Monte Carlo simulation algorithms that generate sparse networks in much less time than conventional

Monte Carlo simulation algorithms. In fact, without the more efficient Monte Carlo simulation algorithms, obtaining bootstrap standard errors would be infeasible.

Finally, while model-based clustering has been limited to networks with fewer than 13,000 nodes and 85 million edge variables [see the largest data set handled to date, Zanghi et al. (2010)], we demonstrate that we can handle much larger, nonbinary networks by considering an internet-based data set with more than 131,000 nodes and 17 billion edge variables, where "edge variables" comprise all observations, including node pairs between which no edge exists. Many internet-based companies and websites, such as http://amazon.com, http://netflix.com and http://epinions.com, allow users to review products and services. Because most users of the World Wide Web do not know each other and thus cannot be sure whether to trust each other, readers of reviews may be interested in an indication of the trustworthiness of the reviewers themselves. A convenient and inexpensive approach is based on evaluations of reviewers by readers. The data set we analyze in Section 7 comes from the website http://epinions.com, which collects such data by allowing any user i to evaluate any other user j as either untrustworthy, coded as $y_{ij} = -1$, or trustworthy, coded as $y_{ij} = +1$, where $y_{ij} = 0$ means that user i did not evaluate user j [Massa and Avesani (2007)]. The resulting network consists of $n = 131{,}827$ users and $N = n(n-1) = 17{,}378{,}226{,}102$ observations. Since each user can only review a relatively small number of other users, the network is sparse: the vast majority of the observations y_{ij} are zero, with only 840,798 negative and positive evaluations. Our modeling goal, broadly speaking, is both to cluster the users based on the patterns of trusts and distrusts in this network and to understand the features of the various clusters by examining model parameters.

The rest of the article is structured as follows: A scalable model-based clustering framework based on finite mixture models is introduced in Section 2. Approximate maximum likelihood and Bayesian estimation are discussed in Sections 3 and 4, respectively, and an algorithm for Monte Carlo simulation of large networks is described in Section 5. Section 6 compares the variational GEM algorithm to the variational EM algorithm of Daudin, Picard and Robin (2008). Section 7 applies our methods to the trust network discussed above.

2. Models for large, discrete-valued networks. We consider n nodes, indexed by integers $1, \ldots, n$, and edges y_{ij} between pairs of nodes i and j, where y_{ij} can take values in a finite set of M elements. By convention, $y_{ii} = 0$ for all i, where 0 signifies "no relationship." We call the set of all edges y_{ij} a discrete-valued network, which we denote by $\mathbf{y}$, and we let $\mathcal{Y}$ denote the set of possible values of $\mathbf{y}$. Special cases of interest are (a) undirected binary networks $\mathbf{y}$, where $y_{ij} \in \{0, 1\}$ is subject to the linear constraint $y_{ij} = y_{ji}$ for all $i < j$; (b) directed binary networks $\mathbf{y}$, where $y_{ij} \in \{0, 1\}$ for all i, j; and (c) directed signed networks $\mathbf{y}$, where $y_{ij} \in \{-1, 0, 1\}$ for all i, j.

A general approach to modeling discrete-valued networks is based on exponential families of distributions [Besag (1974), Frank and Strauss (1986)]:

$$P_{\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y}\mid\mathbf{x})=\exp\bigl[\boldsymbol{\theta}^{\top}\mathbf{g}(\mathbf{x},\mathbf{y})-\psi(\boldsymbol{\theta})\bigr],\qquad \mathbf{y}\in\mathcal{Y}, \tag{2.1}$$

where $\boldsymbol{\theta}$ is the vector of canonical parameters and $\mathbf{g}(\mathbf{x},\mathbf{y})$ is the vector of canonical statistics depending on a matrix $\mathbf{x}$ of covariates, measured on the nodes or the pairs of nodes, and the network $\mathbf{y}$, and $\psi(\boldsymbol{\theta})$ is given by

$$\psi(\boldsymbol{\theta})=\log\sum_{\mathbf{y}'\in\mathcal{Y}}\exp\bigl[\boldsymbol{\theta}^{\top}\mathbf{g}(\mathbf{x},\mathbf{y}')\bigr],\qquad \boldsymbol{\theta}\in\mathbb{R}^{p}, \tag{2.2}$$

and ensures that $P_{\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y}\mid\mathbf{x})$ sums to 1.

A number of exponential family models have been proposed [e.g., Frank and Strauss (1986), Holland and Leinhardt (1981), Hunter and Handcock (2006), Snijders et al. (2006), Wasserman and Pattison (1996)]. In general, though, exponential family models are not scalable: the computing time to evaluate the likelihood function is $\exp(N\log M)$, where $N=n(n-1)/2$ in the case of undirected edges and $N=n(n-1)$ in the case of directed edges, which necessitates time-consuming estimation algorithms [e.g., Caimo and Friel (2011), Hunter and Handcock (2006), Koskinen, Robins and Pattison (2010), Møller et al. (2006), Snijders (2002)].

We therefore restrict attention to scalable exponential family models, which are characterized by dyadic independence:

$$P_{\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y}\mid\mathbf{x})=\prod_{i<j}^{n}P_{\boldsymbol{\theta}}(D_{ij}=d_{ij}\mid\mathbf{x}), \tag{2.3}$$

where $D_{ij}\equiv D_{ij}(\mathbf{Y})$ corresponds to Y_{ij} in the case of undirected edges and (Y_{ij},Y_{ji}) in the case of directed edges. The subscripted $i<j$ and superscripted n mean that the product in (2.3) should be taken over all pairs (i,j) with $1\le i<j\le n$; the same is true for sums as in (3.5).

Dyadic independence has at least three advantages: (a) it facilitates estimation, because the computing time to evaluate the likelihood function scales linearly with N; (b) it facilitates simulation, because dyads are independent; and (c) by design it bypasses the so-called model degeneracy problem: if N is large, some exponential family models without dyadic independence tend to be ill-defined and impractical for modeling networks [Handcock (2003), Schweinberger (2011), Strauss (1986)].

A disadvantage is that most exponential families with dyadic independence are either simplistic [e.g., models with identically distributed edges, Erdős and Rényi (1959), Gilbert (1959)] or nonparsimonious [e.g., the p_1 model with $O(n)$ parameters, Holland and Leinhardt (1981)].

We therefore assume that the probability mass function has a K-component mixture form as follows:

$$
\begin{aligned}
P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y}\mid\mathbf{x}) &= \sum_{\mathbf{z}\in\mathcal{Z}} P_{\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y}\mid\mathbf{x},\mathbf{Z}=\mathbf{z})P_{\boldsymbol{\gamma}}(\mathbf{Z}=\mathbf{z}) \\
&= \sum_{\mathbf{z}\in\mathcal{Z}}\prod_{i<j}^{n} P_{\boldsymbol{\theta}}(D_{ij}=d_{ij}\mid\mathbf{x},\mathbf{Z}=\mathbf{z})P_{\boldsymbol{\gamma}}(\mathbf{Z}=\mathbf{z}),
\end{aligned}
\tag{2.4}
$$

where $\mathbf{Z}$ denotes the membership indicators $\mathbf{Z}_1,\ldots,\mathbf{Z}_n$ with distributions

$$
\mathbf{Z}_i \mid \gamma_1,\ldots,\gamma_K \overset{\text{i.i.d.}}{\sim} \text{Multinomial}(1;\gamma_1,\ldots,\gamma_K) \tag{2.5}
$$

and $\mathcal{Z}$ denotes the support of $\mathbf{Z}$. In some applications, it may be desired to model the membership indicators $\mathbf{Z}_i$ as functions of $\mathbf{x}$ by using multinomial logit or probit models with $\mathbf{Z}_i$ as the outcome variables and $\mathbf{x}$ as predictors [e.g., Tallberg (2005)]. We do not elaborate on such models here, but the variational GEM algorithms discussed in Sections 3 and 4 could be adapted to such models.

Mixture models represent a reasonable compromise between model parsimony and complexity. In particular, the assumption of conditional dyadic independence does *not* imply marginal dyadic independence, which means that the mixture model of (2.4) captures some degree of dependence among the dyads. We give two specific examples of mixture models below.

Example 1. The p_1 model of Holland and Leinhardt (1981) for directed, binary-valued networks may be modified using a mixture model. The original p_1 models the sequence of in-degrees (number of incoming edges of nodes) and out-degrees (number of outgoing edges of nodes) as well as reciprociated edges, postulating that the dyads are independent and that the dyadic probabilities are of the form

$$
P_{\boldsymbol{\theta}}(D_{ij}=d_{ij}) = \exp\bigl[(\alpha_i+\beta_j)y_{ij} + (\alpha_j+\beta_i)y_{ji} + \rho y_{ij}y_{ji} - \psi_{ij}(\boldsymbol{\theta})\bigr], \tag{2.6}
$$

where $\boldsymbol{\theta} = (\alpha_1,\ldots,\alpha_n,\beta_1,\ldots,\beta_n,\rho)$ and $\exp\{-\psi_{ij}(\boldsymbol{\theta})\}$ is a normalizing constant. Following Holland and Leinhardt (1981), the parameters α_i may be interpreted as activity or productivity parameters, representing the tendencies of nodes i to "send" edges to other nodes; the parameters β_j may be interpreted as attractiveness or popularity parameters, representing the tendencies of nodes j to "receive" edges from other nodes; and the parameter ρ may be interpreted as a mutuality or reciprocity parameter, representing the tendency of nodes i and j to reciprocate edges.

A drawback of this model is that it requires $2n+1$ parameters. Here, we show how to extend it to a mixture model that is applicable to both directed and undirected networks as well as discrete-valued networks, that is much more parsimonious, and that allows identification of influential nodes.

Observe that the dyadic probabilities of (2.6) are of the form

$$P_{\boldsymbol{\theta}}(D_{ij} = d_{ij}) \propto \exp\bigl[\boldsymbol{\theta}_1^\top \mathbf{g}_1(d_{ij}) + \boldsymbol{\theta}_{2i}^\top \mathbf{g}_{2i}(d_{ij}) + \boldsymbol{\theta}_{2j}^\top \mathbf{g}_{2j}(d_{ij})\bigr], \tag{2.7}$$

where $\boldsymbol{\theta}_1 = \rho$ is the reciprocity parameter and $\boldsymbol{\theta}_{2i} = (\alpha_i, \beta_i)^\top$ and $\boldsymbol{\theta}_{2j} = (\alpha_j, \beta_j)^\top$ are the sending and receiving propensities of nodes i and j, respectively. The corresponding statistics are the reciprocity indicator $\mathbf{g}_1(d_{ij}) = y_{ij}y_{ji}$ and the sending and receiving indicators $\mathbf{g}_{2i}(d_{ij}) = (y_{ij}, y_{ji})^\top$ and $\mathbf{g}_{2j}(d_{ij}) = (y_{ji}, y_{ij})^\top$ of nodes i and j, respectively. A mixture model modification of the p_1 model postulates that, conditional on $\mathbf{Z}$, the dyadic probabilities are independent and of the form

$$\begin{aligned} &P_{\boldsymbol{\theta}}(D_{ij} = d_{ij} \mid Z_{ik} = Z_{jl} = 1) \\ &\qquad \propto \exp\bigl[\boldsymbol{\theta}_1^\top \mathbf{g}_1(d_{ij}) + \boldsymbol{\theta}_{2k}^\top \mathbf{g}_{2k}(d_{ij}) + \boldsymbol{\theta}_{2l}^\top \mathbf{g}_{2l}(d_{ij})\bigr], \end{aligned} \tag{2.8}$$

where the parameter vectors $\boldsymbol{\theta}_{2k}$ and $\boldsymbol{\theta}_{2l}$ depend on the components k and l to which the nodes i and j belong, respectively. The mixture model version of the p_1 model is therefore much more parsimonious provided $K \ll n$ and was proposed by Schweinberger, Petrescu-Prahova and Vu (2012) in the case of undirected, binary-valued networks. Here, the probabilities of (2.7) and (2.8) are applicable to both undirected and directed networks as well as discrete-valued networks, because the functions $\mathbf{g}_{1k}$ and $\mathbf{g}_{2l}$ may be customized to fit the situation and may even depend on covariates $\mathbf{x}$, though we have suppressed this possibility in the notation. Finally, the mixture model version of the p_1 model admits model-based clustering of nodes based on indegrees or outdegrees or both. A small number of nodes with high indegree or outdegree or both is considered to be influential: if the corresponding nodes were to be removed, the network structure would be impacted.

Example 2. The mixture model of Nowicki and Snijders (2001) assumes that, conditional on $\mathbf{Z}$, the dyads are independent and the conditional dyadic probabilities are of the form

$$P_{\boldsymbol{\pi}}(D_{ij} = d \mid Z_{ik} = Z_{jl} = 1) = \pi_{d;kl}. \tag{2.9}$$

In other words, conditional on $\mathbf{Z}$, the dyad probabilities are constant across dyads and do not depend on covariates. It is straightforward to add covariates by writing the conditional dyad probabilities in canonical form:

$$P_{\boldsymbol{\theta}}(D_{ij} = d_{ij} \mid \mathbf{x}, Z_{ik} = Z_{jl} = 1) \propto \exp\bigl[\boldsymbol{\theta}_1^\top \mathbf{g}_1(\mathbf{x}, d_{ij}) + \boldsymbol{\theta}_{kl}^\top \mathbf{g}_2(\mathbf{x}, d_{ij})\bigr], \tag{2.10}$$

where the canonical statistic vectors $\mathbf{g}_1(\mathbf{x}, d_{ij})$ and $\mathbf{g}_2(\mathbf{x}, d_{ij})$ may depend on the covariates $\mathbf{x}$. If the canonical parameter vectors $\boldsymbol{\theta}_{kl}$ are constrained by the linear constraints $\boldsymbol{\theta}_{kl} = \boldsymbol{\theta}_k + \boldsymbol{\theta}_l$, where $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_l$ are parameter vectors of the same dimension as $\boldsymbol{\theta}_{kl}$, then the mixture model version of the p_1 model arises. In other words, the mixture model version of the p_1 model can be viewed as a constrained version of the Nowicki and Snijders (2001) model. While the constrained version

can be used to cluster nodes based on degree, the unconstrained version can be used to identify, for instance, high-density regions of the network, corresponding to subsets of nodes with large numbers of within-subset edges. These regions may then be studied individually in more detail by using more advanced statistical models such as exponential family models without dyadic independence as proposed by, for example, Holland and Leinhardt (1981), Frank and Strauss (1986), Strauss and Ikeda (1990), Wasserman and Pattison (1996), Snijders et al. (2006) or Hunter and Handcock (2006).

Other examples. Other mixture models for networks have been proposed by Tallberg (2005), Handcock, Raftery and Tantrum (2007) and Airoldi et al. (2008). However, these models scale less well to large networks, so we confine attention here to examples 1 and 2.

3. Approximate maximum likelihood estimation. A standard approach to maximum likelihood estimation of finite mixture models is based on the classical EM algorithm, taking the complete data to be $(\mathbf{Y}, \mathbf{Z})$, where $\mathbf{Z}$ is unobserved [Dempster, Laird and Rubin (1977)]. However, the E-step of an EM algorithm requires the computation of the conditional expectation of the complete data log-likelihood function under the distribution of $\mathbf{Z} \mid \mathbf{Y}$, which is intractable here even in the simplest cases [Daudin, Picard and Robin (2008)].

As an alternative, we consider so-called variational EM algorithms, which can be considered as generalizations of EM algorithms. The basic idea of variational EM algorithms is to construct a tractable lower bound on the intractable log-likelihood function and maximize the lower bound, yielding approximate maximum likelihood estimates. Celisse, Daudin and Pierre (2011) have shown that approximate maximum likelihood estimators along these lines are—at least in the absence of parameter constraints—consistent estimators.

We assume that all modeling of $\mathbf{Y}$ can be conditional on covariates $\mathbf{x}$ and define

$$\pi_{d;ij,kl,\mathbf{x}}(\boldsymbol{\theta}) = P_{\boldsymbol{\theta}}(D_{ij} = d \mid Z_{ik} = Z_{jl} = 1, \mathbf{x}).$$

However, for ease of presentation, we drop the notational dependence of $\pi_{d;ij,kl,\mathbf{x}}$ on i, j, $\mathbf{x}$ and make the homogeneity assumption

$$\pi_{d;ij,kl,\mathbf{x}}(\boldsymbol{\theta}) = \pi_{d;kl}(\boldsymbol{\theta}) \qquad \text{for all } i, j, \mathbf{x}, \tag{3.1}$$

which is satisfied by the models in examples 1 and 2. Exponential parameterizations of $\pi_{d;kl}(\boldsymbol{\theta})$, as in (2.6) and (2.10), may or may not be convenient. An attractive property of the variational EM algorithm proposed here is that it can handle all possible parameterizations of $\pi_{d;kl}(\boldsymbol{\theta})$. In some cases (e.g., example 1), exponential parameterizations are more advantageous than others, while in other cases (e.g., example 2), the reverse holds.

3.1. *Variational EM algorithm*. Let $A(\mathbf{z}) \equiv P(\mathbf{Z} = \mathbf{z})$ be an auxiliary distribution with support $\mathcal{Z}$. Using Jensen's inequality, the log-likelihood function can

be bounded below as follows:

$$\begin{aligned}\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y}) &= \log \sum_{\mathbf{z}\in\mathcal{Z}} \frac{P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})}{A(\mathbf{z})} A(\mathbf{z}) \\ &\geq \sum_{\mathbf{z}\in\mathcal{Z}} \left[\log \frac{P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})}{A(\mathbf{z})}\right] A(\mathbf{z}) \qquad (3.2)\\ &= E_A\big[\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})\big] - E_A\big[\log A(\mathbf{Z})\big].\end{aligned}$$

Some choices of $A(\mathbf{z})$ give rise to better lower bounds than others. To see which choice gives rise to the best lower bound, observe that the difference between the log-likelihood function and the lower bound is equal to the Kullback–Leibler divergence from $A(\mathbf{z})$ to $P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Z}=\mathbf{z}\mid\mathbf{Y}=\mathbf{y})$:

$$\begin{aligned}&\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y}) - \sum_{\mathbf{z}\in\mathcal{Z}} \left[\log \frac{P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})}{A(\mathbf{z})}\right] A(\mathbf{z}) \\ &= \sum_{\mathbf{z}\in\mathcal{Z}} \big[\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y})\big] A(\mathbf{z}) - \sum_{\mathbf{z}\in\mathcal{Z}} \left[\log \frac{P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})}{A(\mathbf{z})}\right] A(\mathbf{z}) \qquad (3.3)\\ &= \sum_{\mathbf{z}\in\mathcal{Z}} \left[\log \frac{A(\mathbf{z})}{P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Z}=\mathbf{z}\mid\mathbf{Y}=\mathbf{y})}\right] A(\mathbf{z}).\end{aligned}$$

If the choice of $A(\mathbf{z})$ were unconstrained in the sense that we could choose from the set of all distributions with support $\mathcal{Z}$, then the best lower bound is obtained by the choice $A(\mathbf{z}) = P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Z}=\mathbf{z}\mid\mathbf{Y}=\mathbf{y})$, which reduces the Kullback–Leibler divergence to 0 and makes the lower bound tight. If the optimal choice is intractable, as is the case here, then it is convenient to constrain the choice to a subset of tractable choices and substitute a choice which, within the subset of tractable choices, is as close as possible to the optimal choice in terms of Kullback–Leibler divergence. A natural subset of tractable choices is given by introducing the auxiliary parameters $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n)$ and setting

$$A(\mathbf{z}) = P_{\boldsymbol{\alpha}}(\mathbf{Z}=\mathbf{z}) = \prod_{i=1}^{n} P_{\boldsymbol{\alpha}_i}(\mathbf{Z}_i=\mathbf{z}_i), \qquad (3.4)$$

where the marginal auxiliary distributions $P_{\boldsymbol{\alpha}_i}(\mathbf{Z}_i=\mathbf{z}_i)$ are Multinomial$(1; \alpha_{i1}, \ldots, \alpha_{iK})$. In this case, the lower bound may be written

$$\begin{aligned}\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma},\boldsymbol{\theta};\boldsymbol{\alpha}) &= E_{\boldsymbol{\alpha}}\big[\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})\big] - E_{\boldsymbol{\alpha}}\big[\log P_{\boldsymbol{\alpha}}(\mathbf{Z})\big] \\ &= \sum_{i<j}^{n}\sum_{k=1}^{K}\sum_{l=1}^{K} \alpha_{ik}\alpha_{jl} \log \pi_{d_{ij};kl}(\boldsymbol{\theta}) \qquad (3.5)\\ &\quad + \sum_{i=1}^{n}\sum_{k=1}^{K} \alpha_{ik}(\log\gamma_k - \log\alpha_{ik}).\end{aligned}$$

Because equation (3.4) assumes independence, the Kullback–Leibler divergence between $P_{\boldsymbol{\alpha}}(\mathbf{Z} = \mathbf{z})$ and $P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Z} = \mathbf{z} \mid \mathbf{Y} = \mathbf{y})$, and thus the tightness of the lower bound, is determined by the dependence of the random variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ conditional on $\mathbf{Y}$. If the random variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ are independent conditional on $\mathbf{Y}$, then, for each i, there exists $\boldsymbol{\alpha}_i$ such that $P_{\boldsymbol{\alpha}_i}(\mathbf{Z}_i = \mathbf{z}_i) = P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Z}_i = \mathbf{z}_i \mid \mathbf{Y} = \mathbf{y})$, which reduces the Kullback–Leibler divergence to 0 and makes the lower bound tight. In general, the random variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ are not independent conditional on $\mathbf{Y}$ and the Kullback–Leibler divergence (3.3) is thus positive.

Approximate maximum likelihood estimates of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ can be obtained by maximizing the lower bound in (3.5) using variational EM algorithms of the following form, where t is the iteration number:

E-STEP: Letting $\boldsymbol{\gamma}^{(t)}$ and $\boldsymbol{\theta}^{(t)}$ denote the current values of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$, maximize $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$. Let $\boldsymbol{\alpha}^{(t+1)}$ denote the optimal value of $\boldsymbol{\alpha}$ and compute $E_{\boldsymbol{\alpha}^{(t+1)}}[\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z})]$.

M-STEP: Maximize $E_{\boldsymbol{\alpha}^{(t+1)}}[\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z})]$ with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$, which is equivalent to maximizing $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}^{(t+1)})$ with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$.

The method ensures that the lower bound is nondecreasing in the iteration number:

$$\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t)}) \le \mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t+1)}) \tag{3.6}$$

$$\le \mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\theta}^{(t+1)}; \boldsymbol{\alpha}^{(t+1)}), \tag{3.7}$$

where inequalities (3.6) and (3.7) follow from the E-step and M-step, respectively.

It is instructive to compare the variational EM algorithm to the classical EM algorithm as applied to finite mixture models. The E-step of the variational EM algorithm minimizes the Kullback–Leibler divergence between $A(\mathbf{z})$ and $P_{\boldsymbol{\gamma}^{(t)},\boldsymbol{\theta}^{(t)}}(\mathbf{Z} = \mathbf{z} \mid \mathbf{Y} = \mathbf{y})$. If the choice of $A(\mathbf{z})$ were unconstrained, then the optimal choice would be $A(\mathbf{z}) = P_{\boldsymbol{\gamma}^{(t)},\boldsymbol{\theta}^{(t)}}(\mathbf{Z} = \mathbf{z} \mid \mathbf{Y} = \mathbf{y})$. Therefore, in the unconstrained case, the E-step of the variational EM algorithm reduces to the E-step of the classical EM algorithm, so the classical EM algorithm can be considered to be the optimal variational EM algorithm.

3.1.1. *Generalized E-step*: *An MM algorithm.* To implement the E-step, we exploit the fact that the lower bound is nondecreasing as long as the E-step and M-step increase the lower bound. In other words, we do not need to maximize the lower bound in the E-step and M-step. Indeed, increasing rather than maximizing the lower bound in the E-step and M-step may have computational advantages when n is large. In the literature on EM algorithms, the advantages of incremental E-steps and incremental M-steps are discussed by Neal and Hinton (1993) and Dempster, Laird and Rubin (1977), respectively. We refer to the variational EM algorithm with either an incremental E-step or an incremental M-step or both as a variational generalized EM, or variational GEM, algorithm.

Direct maximization of $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha})$ is unattractive: equation (3.5) shows that the lower bound depends on the products $\alpha_{ik}\alpha_{jl}$ and, therefore, fixed-point updates of α_{ik} along the lines of [Daudin, Picard and Robin (2008)] depend on all other α_{jl}. We demonstrate in Section 6 that the variational EM algorithm with the fixed-point implementation of the E-step can be inferior to the variational GEM algorithm when K is large.

To separate the parameters of the maximization problem, we increase $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha})$ via an MM algorithm [Hunter and Lange (2004)]. MM algorithms can be viewed as generalizations of EM algorithms [Hunter and Lange (2004)] and are based on iteratively constructing and then optimizing surrogate (minorizing) functions to facilitate the maximization problem in certain situations. We consider here the surrogate function

$$
\begin{aligned}
Q_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t)}, \boldsymbol{\alpha}) = {} & \sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \left(\alpha_{ik}^2 \frac{\alpha_{jl}^{(t)}}{2\alpha_{ik}^{(t)}} + \alpha_{jl}^2 \frac{\alpha_{ik}^{(t)}}{2\alpha_{jl}^{(t)}} \right) \log \pi_{d_{ij};kl}(\boldsymbol{\theta}^{(t)}) \\
& + \sum_{i=1}^{n} \sum_{k=1}^{K} \alpha_{ik} \left(\log \gamma_k^{(t)} - \log \alpha_{ik}^{(t)} - \frac{\alpha_{ik}}{\alpha_{ik}^{(t)}} + 1 \right),
\end{aligned} \tag{3.8}
$$

which we show in Appendix A to have the following two properties:

$$
Q_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\alpha}^{(t)}; \boldsymbol{\alpha}) \le \mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}) \qquad \text{for all } \boldsymbol{\alpha}, \tag{3.9}
$$

$$
Q_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\alpha}^{(t)}; \boldsymbol{\alpha}^{(t)}) = \mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t)}). \tag{3.10}
$$

In the language of MM algorithms, conditions (3.9) and (3.10) establish that $Q_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\alpha}^{(t)}; \boldsymbol{\alpha})$ is a *minorizer* of $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha})$ at $\boldsymbol{\alpha}^{(t)}$. The theory of MM algorithms implies that maximizing the minorizer with respect to $\boldsymbol{\alpha}$ forces $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha})$ uphill [Hunter and Lange (2004)]. This maximization, involving n separate quadratic programming problems of K variables $\boldsymbol{\alpha}_i$ under the constraints $\alpha_{ik} \ge 0$ for all k and $\sum_{k=1}^{K} \alpha_{ik} = 1$, may be accomplished quickly using the method described by Stefanov (2004). When n is large, it is much easier to update $\boldsymbol{\alpha}$ by maximizing the Q_{ML} function, which is the sum of functions of the individual $\boldsymbol{\alpha}_i$, than by maximizing the $\mathrm{LB}_{\mathrm{ML}}$ function, in which the $\boldsymbol{\alpha}$ parameters are not separated in this way. We therefore arrive at the following replacement for the E-step:

GENERALIZED E-STEP: For $i = 1, \ldots, n$, increase $Q_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\alpha}^{(t)}; \boldsymbol{\alpha})$ as a function of $\boldsymbol{\alpha}_i$ subject to $\alpha_{ik} \ge 0$ for all k and $\sum_{k=1}^{K} \alpha_{ik} = 1$. Let $\boldsymbol{\alpha}^{(t+1)}$ denote the new value of $\boldsymbol{\alpha}$.

3.1.2. *More on the M-step.* To maximize $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}^{(t+1)})$ in the M-step, examination of (3.5) shows that maximization with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ may be

accomplished separately. In fact, for $\boldsymbol{\gamma}$, there is a simple, closed-form solution:

$$\gamma_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} \alpha_{ik}^{(t+1)}, \qquad k = 1, \ldots, K. \tag{3.11}$$

Concerning $\boldsymbol{\theta}$, if there are no constraints on $\boldsymbol{\pi}(\boldsymbol{\theta})$ other than $\sum_{d\in\mathcal{D}} \pi_{d;kl}(\boldsymbol{\theta}) = 1$, it is preferable to maximize with respect to $\boldsymbol{\pi} = \boldsymbol{\pi}(\boldsymbol{\theta})$ rather than $\boldsymbol{\theta}$, because there are closed-form expressions for $\boldsymbol{\pi}^{(t+1)}$ but not for $\boldsymbol{\theta}^{(t+1)}$. Maximization with respect to $\boldsymbol{\pi}$ is accomplished by setting

$$\pi_{d;kl}^{(t+1)} = \frac{\sum_{i<j}^{n} \alpha_{ik}^{(t+1)} \alpha_{jl}^{(t+1)} I(D_{ij} = d)}{\sum_{i<j}^{n} \alpha_{ik}^{(t+1)} \alpha_{jl}^{(t+1)}}, \qquad d \in \mathcal{D}, k, l = 1, \ldots, K. \tag{3.12}$$

If the homogeneity assumption (3.1) does not hold, then closed-form expressions for $\boldsymbol{\pi}$ may not be available. In some cases, as in the presence of categorical covariates, closed form expressions for $\boldsymbol{\pi}$ are available, but the dimension of $\boldsymbol{\pi}$, and thus computing time, increases with the number of categories.

If equations (2.1) and (2.3) hold, then the exponential parametrization $\boldsymbol{\pi}(\boldsymbol{\theta})$ may be inverted to obtain an approximate maximum likelihood estimate of $\boldsymbol{\theta}$ after the approximate MLE of $\boldsymbol{\pi}$ is found using the variational GEM algorithm. One method for accomplishing this inversion exploits the convex duality of exponential families [Barndorff-Nielsen (1978), Wainwright and Jordan (2008)] and is explained in Appendix B.

If, in addition to the constraint $\sum_{d\in\mathcal{D}} \pi_{d;kl}(\boldsymbol{\theta}) = 1$, additional constraints on $\boldsymbol{\pi}$ are present, the maximization with respect to $\boldsymbol{\pi}$ may either decrease or increase computing time. Linear constraints on $\boldsymbol{\pi}$ can be enforced by Lagrange multipliers and reduce the dimension of $\boldsymbol{\pi}$ and thus computing time. Nonlinear constraints on $\boldsymbol{\pi}$, as in example 1, may not admit closed form updates of $\boldsymbol{\pi}$ and thus may require iterative methods. If so, and if the nonlinear constraints stem from exponential family parameterizations of $\boldsymbol{\pi}(\boldsymbol{\theta})$ with natural parameter vector $\boldsymbol{\theta}$ as in example 1, then it is convenient to translate the constrained maximization problem into an unconstrained problem by maximizing $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}^{(t+1)})$ with respect to $\boldsymbol{\theta}$ and exploiting the fact that $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}^{(t+1)})$ is a concave function of $\boldsymbol{\theta}$ owing to the exponential family membership of $\pi_{d;kl}(\boldsymbol{\theta})$ [Barndorff-Nielsen (1978), page 150]. We show in Appendix C how the exponential family parameterization can be used to derive the gradient and Hessian of the lower bound of $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}^{(t+1)})$ with respect to $\boldsymbol{\theta}$, which we exploit in Section 7 using a Newton–Raphson algorithm.

3.2. *Standard errors.* Although we maximize the lower bound $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ of the log-likelihood function to obtain approximate maximum likelihood estimates, standard errors of the approximate maximum likelihood estimates $\hat{\boldsymbol{\gamma}}$ and $\hat{\boldsymbol{\theta}}$ based on the curvature of the lower bound $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ may be too small. The

reason is that even when the lower bound is close to the log-likelihood function, the lower bound may be more curved than the log-likelihood function [Wang and Titterington (2005)]; indeed, the higher curvature helps ensure that $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ is a lower bound of the log-likelihood function $\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y} = \mathbf{y})$ in the first place. As an alternative, we approximate the standard errors of the approximate maximum likelihood estimates of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ by a parametric bootstrap method [Efron (1979)] that can be described as follows:

(1) Given the approximate maximum likelihood estimates of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$, sample B data sets.

(2) For each data set, compute the approximate maximum likelihood estimates of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$.

In addition to fast maximum likelihood algorithms, the parametric bootstrap method requires fast simulation algorithms. We propose such an algorithm in Section 5.

3.3. *Starting and stopping*. As usual with EM-like algorithms, it is a good idea to use multiple different starting values with the variational EM due to the existence of distinct local maxima. We find it easiest to use random starts in which we assign the values of $\boldsymbol{\alpha}^{(0)}$ and then commence with an M-step. This results in values $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\theta}^{(0)}$, then the algorithm continues with the first E-step, and so on. The initial $\alpha_{ik}^{(0)}$ are chosen independently uniformly randomly on $(0, 1)$, then each $\boldsymbol{\alpha}_i^{(0)}$ is multiplied by a normalizing constant chosen so that the elements of $\boldsymbol{\alpha}_i^{(0)}$ sum to one for every i.

The numerical experiments of Section 7 use 100 random restarts each. Ideally, more restarts would be used, yet the size of the data sets with which we work makes every run somewhat expensive. We chose the number 100 because we were able to parallelize on a fairly large scale, essentially running 100 separate copies of the algorithm. Larger numbers of runs, such as 1000, would have forced longer run times since we would have had to run some of the trials in series rather than in parallel.

As a convergence criterion, we stop the algorithm as soon as

$$\frac{|\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\theta}^{(t+1)}; \boldsymbol{\alpha}^{(t+1)}) - \mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t)})|}{|\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\theta}^{(t+1)}; \boldsymbol{\alpha}^{(t+1)})|} < 10^{-10}.$$

We consider the relative change in the objective function rather than the absolute change or the changes in the parameters themselves because (1) even small changes in the parameter values can result in large changes of the objective function, and (2) the objective function is a lower bound of the log-likelihood, so small absolute changes of the objective function may not be worth the computational effort.

4. Approximate Bayesian estimation. The key to Bayesian model estimation and model selection is the marginal likelihood, defined as

$$P(\mathbf{Y}=\mathbf{y}) = \int_{\Gamma}\int_{\Theta}\sum_{\mathbf{z}\in\mathcal{Z}} P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})\,p(\boldsymbol{\gamma},\boldsymbol{\theta})\,d\boldsymbol{\gamma}\,d\boldsymbol{\theta}, \tag{4.1}$$

where $p(\boldsymbol{\gamma},\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$. To ensure that the marginal likelihood is well-defined, we assume that the prior distribution is proper, which is common practice in mixture modeling [McLachlan and Peel (2000), Chapter 4]. A lower bound on the log marginal likelihood can be derived by introducing an auxiliary distribution with support $\mathcal{Z}\times\Gamma\times\Theta$, where Γ is the parameter space of $\boldsymbol{\gamma}$ and Θ is the parameter space of $\boldsymbol{\theta}$. A natural choice of auxiliary distributions is given by

$$A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta}) \equiv \Bigg[\prod_{i=1}^{n} P_{\boldsymbol{\alpha}_{\mathbf{Z},i}}(\mathbf{Z}_i=\mathbf{z}_i)\Bigg] p_{\boldsymbol{\alpha}_{\boldsymbol{\gamma}}}(\boldsymbol{\gamma}) \Bigg[\prod_{i=1}^{L} p_{\boldsymbol{\alpha}_{\boldsymbol{\theta}}}(\theta_i)\Bigg], \tag{4.2}$$

where $\boldsymbol{\alpha}$ denotes the set of auxiliary parameters $\boldsymbol{\alpha}_{\mathbf{Z}}=(\boldsymbol{\alpha}_{\mathbf{Z},1},\ldots,\boldsymbol{\alpha}_{\mathbf{Z},n})$, $\boldsymbol{\alpha}_{\boldsymbol{\gamma}}$ and $\boldsymbol{\alpha}_{\boldsymbol{\theta}}$.

A lower bound on the log marginal likelihood can be derived by Jensen's inequality:

$$\begin{aligned}\log P(\mathbf{Y}=\mathbf{y}) &= \log\int_{\Gamma}\int_{\Theta}\sum_{\mathbf{z}\in\mathcal{Z}} \frac{P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})\,p(\boldsymbol{\gamma},\boldsymbol{\theta})}{A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})} A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})\,d\boldsymbol{\gamma}\,d\boldsymbol{\theta} \\ &\geq E_{\boldsymbol{\alpha}}\big[\log P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})\,p(\boldsymbol{\gamma},\boldsymbol{\theta})\big] - E_{\boldsymbol{\alpha}}\big[\log A_{\boldsymbol{\alpha}}(\mathbf{Z},\boldsymbol{\gamma},\boldsymbol{\theta})\big],\end{aligned} \tag{4.3}$$

where the expectations are taken with respect to the auxiliary distribution $A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})$.

We denote the right-hand side of (4.3) by $\mathrm{LB}_B(\boldsymbol{\alpha}_{\boldsymbol{\gamma}},\boldsymbol{\alpha}_{\boldsymbol{\theta}};\boldsymbol{\alpha}_{\mathbf{Z}})$. By an argument along the lines of (3.3), one can show that the difference between the log marginal likelihood and $\mathrm{LB}_B(\boldsymbol{\alpha}_{\boldsymbol{\gamma}},\boldsymbol{\alpha}_{\boldsymbol{\theta}};\boldsymbol{\alpha}_{\mathbf{Z}})$ is equal to the Kullback–Leibler divergence from the auxiliary distribution $A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})$ to the posterior distribution $P(\mathbf{Z}=\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta}\mid\mathbf{Y}=\mathbf{y})$:

$$\begin{aligned}&\log P(\mathbf{Y}=\mathbf{y}) - \int_{\Gamma}\int_{\Theta}\sum_{\mathbf{z}\in\mathcal{Z}}\Bigg[\log\frac{P_{\boldsymbol{\gamma},\boldsymbol{\theta}}(\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z})\,p(\boldsymbol{\gamma},\boldsymbol{\theta})}{A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})}\Bigg] A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})\,d\boldsymbol{\gamma}\,d\boldsymbol{\theta} \\ &\quad= \int_{\Gamma}\int_{\Theta}\sum_{\mathbf{z}\in\mathcal{Z}}\Bigg[\log\frac{A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})}{P(\mathbf{Z}=\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta}\mid\mathbf{Y}=\mathbf{y})}\Bigg] A_{\boldsymbol{\alpha}}(\mathbf{z},\boldsymbol{\gamma},\boldsymbol{\theta})\,d\boldsymbol{\gamma}\,d\boldsymbol{\theta}.\end{aligned} \tag{4.4}$$

The Kullback–Leibler divergence between the auxiliary distribution and the posterior distribution can be minimized by a variational GEM algorithm as follows, where t is the iteration number:

GENERALIZED E-STEP: Letting $\boldsymbol{\alpha}_{\boldsymbol{\gamma}}^{(t)}$ and $\boldsymbol{\alpha}_{\boldsymbol{\theta}}^{(t)}$ denote the current values of $\boldsymbol{\alpha}_{\boldsymbol{\gamma}}$ and $\boldsymbol{\alpha}_{\boldsymbol{\theta}}$, increase $\mathrm{LB}_B(\boldsymbol{\alpha}_{\boldsymbol{\gamma}}^{(t)},\boldsymbol{\alpha}_{\boldsymbol{\theta}}^{(t)};\boldsymbol{\alpha}_{\mathbf{Z}})$ with respect to $\boldsymbol{\alpha}_{\mathbf{Z}}$. Let $\boldsymbol{\alpha}_{\mathbf{Z}}^{(t+1)}$ denote the new value of $\boldsymbol{\alpha}_{\mathbf{Z}}$.

GENERALIZED M-STEP: Choose new values $\boldsymbol{\alpha}_{\boldsymbol{\gamma}}^{(t+1)}$ and $\boldsymbol{\alpha}_{\boldsymbol{\theta}}^{(t+1)}$ that increase $\mathrm{LB}_B(\boldsymbol{\alpha}_{\boldsymbol{\gamma}}, \boldsymbol{\alpha}_{\boldsymbol{\theta}}; \boldsymbol{\alpha}_{\mathbf{Z}}^{(t+1)})$ with respect to $\boldsymbol{\alpha}_{\boldsymbol{\gamma}}$ and $\boldsymbol{\alpha}_{\boldsymbol{\theta}}$.

By construction, iteration t of a variational GEM algorithm increases the lower bound $\mathrm{LB}_B(\boldsymbol{\alpha}_{\boldsymbol{\gamma}}, \boldsymbol{\alpha}_{\boldsymbol{\theta}}; \boldsymbol{\alpha}_{\mathbf{Z}})$:

$$\mathrm{LB}_B\bigl(\boldsymbol{\alpha}_{\boldsymbol{\gamma}}^{(t)}, \boldsymbol{\alpha}_{\boldsymbol{\theta}}^{(t)}; \boldsymbol{\alpha}_{\mathbf{Z}}^{(t)}\bigr) \le \mathrm{LB}_B\bigl(\boldsymbol{\alpha}_{\boldsymbol{\gamma}}^{(t)}, \boldsymbol{\alpha}_{\boldsymbol{\theta}}^{(t)}; \boldsymbol{\alpha}_{\mathbf{Z}}^{(t+1)}\bigr) \tag{4.5}$$

$$\le \mathrm{LB}_B\bigl(\boldsymbol{\alpha}_{\boldsymbol{\gamma}}^{(t+1)}, \boldsymbol{\alpha}_{\boldsymbol{\theta}}^{(t+1)}; \boldsymbol{\alpha}_{\mathbf{Z}}^{(t+1)}\bigr). \tag{4.6}$$

A variational GEM algorithm approximates the marginal likelihood as well as the posterior distribution. Therefore, it tackles Bayesian model estimation and model selection at the same time.

Variational GEM algorithms for approximate Bayesian inference are only slightly more complicated to implement than the variational GEM algorithms for approximate maximum likelihood estimation presented in Section 3. To understand the difference, we examine the analogue of (3.5):

$$\begin{aligned}&\mathrm{LB}_B(\boldsymbol{\alpha}_{\boldsymbol{\gamma}}, \boldsymbol{\alpha}_{\boldsymbol{\theta}}; \boldsymbol{\alpha}_{\mathbf{Z}})\\ &\quad= \sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_{\mathbf{Z},ik} \alpha_{\mathbf{Z},jl} E_{\boldsymbol{\alpha}}\bigl[\log \pi_{d_{ij};kl}(\boldsymbol{\theta})\bigr] + E_{\boldsymbol{\alpha}}\bigl[\log P_{\boldsymbol{\gamma}}(\mathbf{Z}=\mathbf{z})\bigr]\\ &\qquad+ E_{\boldsymbol{\alpha}}\bigl[\log p(\boldsymbol{\gamma}, \boldsymbol{\theta})\bigr] - E_{\boldsymbol{\alpha}}\bigl[\log A(\mathbf{Z}=\mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\theta})\bigr].\end{aligned} \tag{4.7}$$

If the prior distributions of $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ are given by independent Dirichlet and Gaussian distributions and the auxiliary distributions of $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$, $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ are given by independent Multinomial, Dirichlet and Gaussian distributions, respectively, then the expectations on the right-hand side of (4.7) are tractable, with the possible exception of the expectations $E_{\boldsymbol{\alpha}}[\log \pi_{d;kl}(\boldsymbol{\theta})]$. Under the exponential parameterization

$$\pi_{d;kl}(\boldsymbol{\theta}) = \exp\biggl\{\boldsymbol{\theta}^{\top}\mathbf{g}(d) - \log \sum_{d' \in \mathcal{D}} \exp\bigl[\boldsymbol{\theta}^{\top}\mathbf{g}(d')\bigr]\biggr\}, \tag{4.8}$$

the expectations can be written as

$$E_{\boldsymbol{\alpha}}\bigl[\log \pi_{d;kl}(\boldsymbol{\theta})\bigr] = E_{\boldsymbol{\alpha}}[\boldsymbol{\theta}]^{\top}\mathbf{g}(d) - E_{\boldsymbol{\alpha}}\biggl\{\log \sum_{d' \in \mathcal{D}} \exp\bigl[\boldsymbol{\theta}^{\top}\mathbf{g}(d')\bigr]\biggr\} \tag{4.9}$$

and are intractable. We are not aware of parameterizations under which the expectations are tractable. We therefore use exponential parameterizations and deal with the intractable nature of the resulting expectations by invoking Jensen's inequality:

$$E_{\boldsymbol{\alpha}}\bigl[\log \pi_{d;kl}(\boldsymbol{\theta})\bigr] \ge E_{\boldsymbol{\alpha}}[\boldsymbol{\theta}]^{\top}\mathbf{g}(d) - \log \sum_{d' \in \mathcal{D}} E_{\boldsymbol{\alpha}}\bigl\{\exp\bigl[\boldsymbol{\theta}^{\top}\mathbf{g}(d')\bigr]\bigr\}. \tag{4.10}$$

The right-hand side of (4.10) involves expectations of independent log-normal random variables, which are tractable. We thus obtain a looser, yet tractable, lower

bound by replacing $E_{\boldsymbol{\alpha}}[\log \pi_{d;kl}(\boldsymbol{\theta})]$ in (4.7) by the right-hand side of inequality (4.10).

To save space, we do not address the specific numerical techniques that may be used to implement the variational GEM algorithm here. In short, the generalized E-step is based on an MM algorithm along the lines of Section 3.1.1. In the generalized M-step, numerical gradient-based methods may be used. A detailed treatment of this Bayesian estimation method and its implementation, using a more complicated prior distribution, may be found in Schweinberger, Petrescu-Prahova and Vu (2012); code related to this article is available at http://sites.stat.psu.edu/~dhunter/code/.

5. Monte Carlo simulation. Monte Carlo simulation of large, discrete-valued networks serves at least three purposes:

(a) to generate simulated data to be used in simulation studies;
(b) to approximate standard errors of the approximate maximum likelihood estimates by parametric bootstrap;
(c) to assess model goodness of fit by simulation.

A crude Monte Carlo approach is based on sampling $\mathbf{Z}$ by cycling through all n nodes and sampling $D_{ij} \mid \mathbf{Z}$ by cycling through all $n(n-1)/2$ dyads. However, the running time of such an algorithm is $O(n^2)$, which is too slow to be useful in practice, because each of the goals listed above tends to require numerous simulated data sets.

We propose Monte Carlo simulation algorithms that exploit the fact that discrete-valued networks tend to be sparse in the sense that one element of $\mathcal{D}$ is much more common than all other elements of $\mathcal{D}$. An example is given by directed, binary-valued networks, where $\mathcal{D} = \{(0,0),(0,1),(1,0),(1,1)\}$ is the sample space of dyads and $(0,0) \in \mathcal{D}$ tends to dominate all other elements of $\mathcal{D}$.

Assume there exists an element b of $\mathcal{D}$, called the baseline, that dominates the other elements of $\mathcal{D}$ in the sense that $\pi_{b;kl} \gg 1 - \pi_{b;kl}$ for all k and l. The Monte Carlo simulation algorithm exploiting the sparsity of large, discrete-valued networks can be described as follows:

(1) Sample $\mathbf{Z}$ by sampling $\mathbf{M} \sim \text{Multinomial}(n; \gamma_1, \ldots, \gamma_K)$ and assigning nodes $1, \ldots, M_1$ to component 1, nodes $M_1 + 1, \ldots, M_1 + M_2$ to component 2, etc.
(2) Sample $\mathbf{Y} \mid \mathbf{Z}$ as follows: for each $1 \le k \le l \le K$,
   (a) sample the number of dyads S_{kl} with nonbaseline values, $S_{kl} \sim \text{Binomial}(N_{kl}, 1 - \pi_{b;kl})$, where N_{kl} is the number of pairs of nodes belonging to components k and l;
   (b) sample S_{kl} out of N_{kl} pairs of nodes $i < j$ without replacement;
   (c) for each of the S_{kl} sampled pairs of nodes $i < j$, sample the nonbaseline value D_{ij} according to the probabilities $\pi_{d;kl}/(1 - \pi_{b;kl})$, $d \in \mathcal{D}$, $d \neq b$.

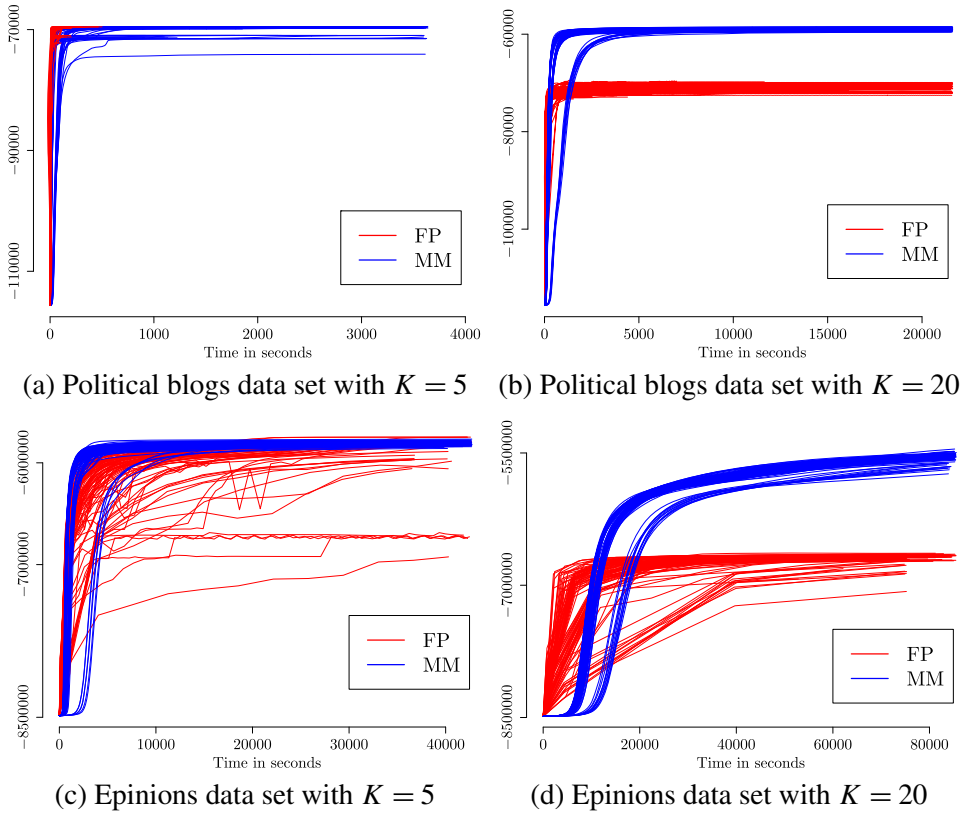
In general, if the degree of any node (i.e., the number of nonbaseline values for all dyad variables incident on that node) has a bounded expectation, then the expected number of nonbaseline values $S = \sum_{k \le l} S_{kl}$ in the network scales with n and the expected running time of the Monte Carlo simulation algorithm scales with $nK^2|\mathcal{D}|$. If K is small and n is large, then the Monte Carlo approach that exploits the sparsity of large, discrete-valued networks is superior to the crude Monte Carlo approach.

6. Comparison of algorithms. We compare the variational EM algorithm based on the fixed-point (FP) implementation of the E-step along the lines of Daudin, Picard and Robin (2008) to the variational GEM algorithm based on the MM implementation of the E-step by applying them to two data sets. The first data set comes from the study on political blogs by Adamic and Glance (2005). We convert the binary network of political blogs with two labels, liberal (+1) and conservative (−1), into a signed network by assigning labels of receivers to the corresponding directed edges. The resulting network has 1490 nodes and 2,218,610 edge variables. The second data set is the Epinions data set described in Section 1 with more than 131,000 nodes and more than 17 billion edge variables.

We compare the two algorithms using the unconstrained network mixture model of (2.9) with $K = 5$ and $K = 20$ components. For the first data set, we allow up to 1 hour for $K = 5$ components and up to 6 hours for $K = 20$ components. For the second data set, we allow up to 12 hours for $K = 5$ components and up to 24 hours for $K = 20$ components. For each data set, for each number of components and for each algorithm, we carried out 100 runs using random starting values as described in Section 3.3.

Figure 1 shows trace plots of the lower bound $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t)})$ of the log-likelihood function, where red lines refer to the lower bound of the variational EM algorithm with FP implementation and blue lines refer to the lower bound of the variational GEM algorithm with MM implementation. The variational EM algorithm seems to outperform the variational GEM algorithm in terms of computing time when K and n are small. However, when K or n are large, the variational GEM algorithm appears far superior to the variational EM algorithm in terms of the lower bounds. The contrast is most striking when K is large, though the variational GEM seems to outperform the variational EM algorithm even when K is small and n is large. We believe that the superior performance of the variational GEM algorithm stems from the fact that it separates the parameters of the maximization problem and reduces the dependence of the updates of the variational parameters α_{ik}, as discussed in Section 3.1.1, while the variational EM algorithm tends to be trapped in local maxima.

Thus, if K and n are small and a computing cluster is available, it seems preferable to carry out a large number of runs using the variational EM algorithm in parallel, using random starting values as described in Section 3.3. However, if either K or n is large, it is preferable to use the variational GEM algorithm. Since



(a) Political blogs data set with $K = 5$ (b) Political blogs data set with $K = 20$

(c) Epinions data set with $K = 5$ (d) Epinions data set with $K = 20$

FIG. 1. *Trace plots of the lower bound* $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t)})$ *of the log-likelihood function for* 100 *runs each of the variational EM algorithm with FP implementation* (*red*) *and variational GEM algorithm with MM implementation* (*blue*), *applied to the unconstrained network mixture model of* (2.9) *for two different data sets*.

the variational GEM algorithm is not prone to be trapped in local maxima, a small number of long runs may be all that is needed.

7. Application. Here, we address the problem of clustering the $n = 131{,}000$ users of the data set introduced in Section 1 according to their levels of trustworthiness, as indicated by the network of $+1$ and -1 ratings given by fellow users. To this end, we first introduce the individual "excess trust" statistics

$$e_i(\mathbf{y}) = \sum_{1 \le j \le n,\, j \ne i} y_{ji}.$$

Since $e_i(\mathbf{y})$ is the number of positive ratings received by user i in excess of the number of negative ratings, it is a natural measure of a user's individual trustworthiness. Our contention is that consideration of the overall pattern of network

connections results in a more revealing clustering pattern than a mere consideration of the $e_i(\mathbf{y})$ statistics, and we support this claim by considering three different clustering methods: A parsimonious network model using the $e_i(\mathbf{y})$ statistics, the fully unconstrained network model of (2.9), and a mixture model that considers only the $e_i(\mathbf{y})$ statistics while ignoring the other network structure.

For each method, we assume that the number of categories, K, is five. Partly, this choice is motivated by the fact that formal model selection methods such as the ICL criterion suggested by Daudin, Picard and Robin (2008), which we discuss in Section 9, suggest dozens if not hundreds of categories, which complicate summary and interpretation. Since the reduction of data is the primary task of statistics [Fisher (1922)], we want to keep the number of categories small and follow the standard practice of internet-based companies and websites, such as http://amazon.com and http://netflix.com, which use five categories to classify the trustworthiness of reviewers, sellers and service providers.

Our parsimonious model, which enjoys benefits over the other two alternatives as we shall see, is based on

$$\begin{aligned} &P_{\boldsymbol{\theta}}(D_{ij} = d_{ij} \mid Z_{ik} = Z_{jl} = 1) \\ &\quad \propto \exp\bigl[\theta^{-}\bigl(y_{ij}^{-} + y_{ji}^{-}\bigr) + \theta^{+}\bigl(y_{ij}^{+} + y_{ji}^{+}\bigr) + \theta_k^{\Delta} y_{ji} \\ &\qquad + \theta_l^{\Delta} y_{ij} + \theta^{--} y_{ij}^{-} y_{ji}^{-} + \theta^{++} y_{ij}^{+} y_{ji}^{+}\bigr], \end{aligned} \tag{7.1}$$

where $y_{ij}^{-} = I(y_{ij} = -1)$ and $y_{ij}^{+} = I(y_{ij} = 1)$ are indicators of negative and positive edges, respectively. The parameters in model (7.1) are not identifiable, because $y_{ij} = y_{ij}^{+} - y_{ij}^{-}$ and $y_{ji} = y_{ji}^{+} - y_{ji}^{-}$. We therefore constrain the positive edge parameter θ^{+} to be 0. Model (7.1) assumes in the interest of model parsimony that the propensities to form negative and positive edges and to reciprocate negative and positive edges do not vary across clusters; however, the flexibility afforded by this modeling framework enables us to define cluster-specific parameters for any of these propensities if we wish. The conditional probability mass function of the whole network is given by

$$\begin{aligned} &P_{\boldsymbol{\theta}}(\mathbf{Y} = \mathbf{y} \mid \mathbf{Z} = \mathbf{z}) \\ &\quad \propto \exp\Biggl[\theta^{-} \sum_{i<j}^{n} \bigl(y_{ij}^{-} + y_{ji}^{-}\bigr) + \sum_{k=1}^{K} \theta_k^{\Delta} t_k(\mathbf{y}, \mathbf{z}) \\ &\qquad + \theta^{--} \sum_{i<j}^{n} y_{ij}^{-} y_{ji}^{-} + \theta^{++} \sum_{i<j}^{n} y_{ij}^{+} y_{ji}^{+}\Biggr], \end{aligned} \tag{7.2}$$

where $t_k(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{n} z_{ik} e_i(\mathbf{y})$ is the total excess trust for all nodes in the kth category. The θ_k^{Δ} parameters are therefore measures of the trustworthiness of each of the categories. Furthermore, these parameters are estimated in the presence of—that is, after correcting for—the reciprocity effects as measured by the parameters

θ^{--} and θ^{++}, which summarize the overall tendencies of users to reciprocate negative and positive ratings, respectively. Thus, θ^{--} and θ^{++} may be considered to measure overall tendencies toward *lex talionis* and *quid pro quo* behaviors.

One alternative model we consider is the unconstrained network model obtained from (2.9). With five components, this model comprises four mixing parameters $\lambda_1, \ldots, \lambda_4$ in addition to the $\pi_{d;kl}$ parameters, of which there are 105: there are nine types of dyads d whenever $k \neq l$, contributing $8\binom{5}{2} = 80$ parameters, and six types of dyads d whenever $k = l$, contributing an additional $5(5) = 25$ parameters. Despite the increased flexibility afforded by model (2.9), we view the loss of interpretability due to the large number of parameters as a detriment. Furthermore, more parameters opens up the possibility of overfitting and, as we discuss below, appears to make the lower bound of the log-likelihood function highly multimodal.

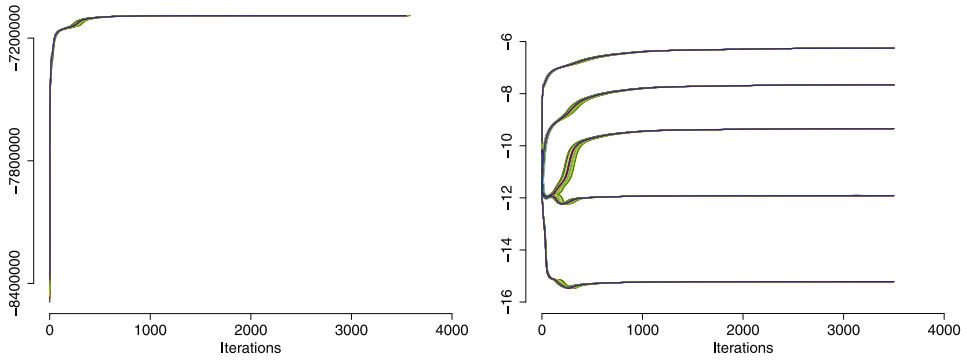
Our other alternative model is a univariate mixture model applied to the $e_i(\mathbf{y})$ statistics directly, which assumes that the individual excesses $e_i(\mathbf{y})$ are independent random variables sampled from a distribution with density

$$f(x) = \sum_{j=1}^{5} \lambda_j \frac{1}{\sigma_j} \phi\left(\frac{x - \mu_j}{\sigma_j}\right), \tag{7.3}$$

where λ_j, μ_j and σ_j are component-specific mixing proportions, means and standard deviations, respectively, and $\phi(\cdot)$ is the standard normal density. Traditional univariate approaches like this are less suitable than network-based clustering approaches not only because by design they neither consider nor inform us about the topology of the network, which may be relevant, but also because the individual excesses are not independent: these $e_i(\mathbf{y})$ are functions of edges, and edges may be dependent owing to reciprocity (and other forms of dependence not modeled here), which decades of research [e.g., Davis (1968), Holland and Leinhardt (1981)] have shown to be important in shaping social networks. Unlike the univariate mixture model of (7.3), the mixture model we employ for networks allows for such dependence.

We use a variational GEM algorithm to estimate the network model (7.2), where the M-step is executed by a Newton–Raphson algorithm using the gradient and Hessian derived in Appendix C with a maximum of 100 iterations. It stops earlier if the largest absolute value in the gradient vector is less than 10^{-10}. By contrast, the unconstrained network model following from (2.9) employs a variational GEM algorithm using the exact M-step update (3.12). The variational GEM algorithm stops when either the relative change in the objective function is less than 10^{-10} or 6000 iterations are performed. Most runs require the full 6000 iterations. To estimate the normal mixture model (7.3), we use the `R` package `mixtools` [Benaglia et al. (2009)].

To diagnose convergence of the algorithm for fitting the model (7.2), we present the trace plot of the lower bound of the log-likelihood function $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)};$



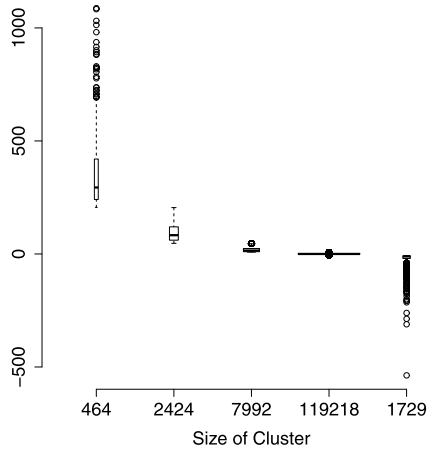
(a) Log-likelihood lower bound (b) Cluster-specific excess parameters

FIG. 2. (a) *Trace plot of the lower bound* $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\theta}^{(t)}; \boldsymbol{\alpha}^{(t)})$ *of the log-likelihood function and* (b) *cluster-specific excess parameters* θ_k^{Δ}, *using* 100 *runs with random starting values.*

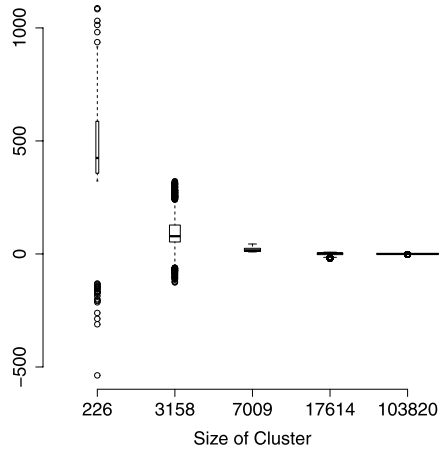
$\boldsymbol{\alpha}^{(t)}$) in Figure 2(a) and the trace plot of the cluster-specific excess parameters θ_k^{Δ} in Figure 2(b). Both figures are based on 100 runs, where the starting values are obtained by the procedure described in Section 3.3. The results suggest that all 100 runs seem to converge to roughly the same solution. This fact is somewhat remarkable, since many variational algorithms appear very sensitive to their starting values, converging to multiple distinct local optima [e.g., Daudin, Pierre and Vacher (2010), Salter-Townshend and Murphy (2013)]. For instance, the 100 runs for the unconstrained network model (2.9) produced essentially a unique set of estimates for each set of random starting values. Similarly, the normal mixture model algorithm produces many different local maxima, even after we try to correct for label-switching by choosing random starting values fairly tightly clustered by their mean values.

Figure 3 shows the observed excesses $e_1(\mathbf{y}), \ldots, e_n(\mathbf{y})$ grouped by clusters for the best solutions, as measured by likelihood or approximate likelihood, found for each of the three clustering methods. It appears that the clustering based on the parsimonious network model does a better job of separating the $e_i(\mathbf{y})$ statistics into distinct subgroups—though this is not the sole criterion used—than the clusterings for the other two models, which are similar to each other. In addition, if we use a normal mixture model in which the variances are restricted to be constant across components, the results are even worse, with one large cluster and multiple clusters with few nodes.

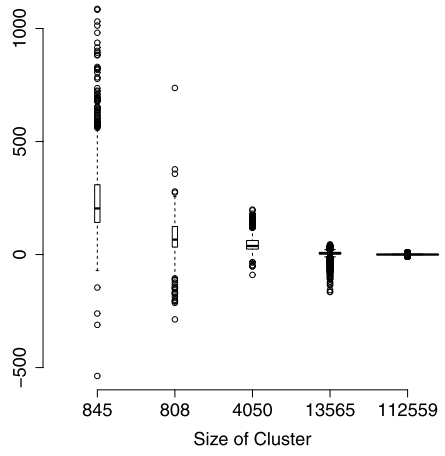
In Figure 4, we "ground truth" the clustering solutions using external information: the average ratings of 659,290 articles, grouped according to the highest-probability category of the article's author. While in Figure 3 the size of each cluster is the number of users in that cluster, in Figure 4 the size of each cluster is the number of articles written by users in that cluster. The widths of the boxes in Figures 3 and 4 are proportional to the square roots of the cluster sizes.



(a) Network mixture model (7.2)



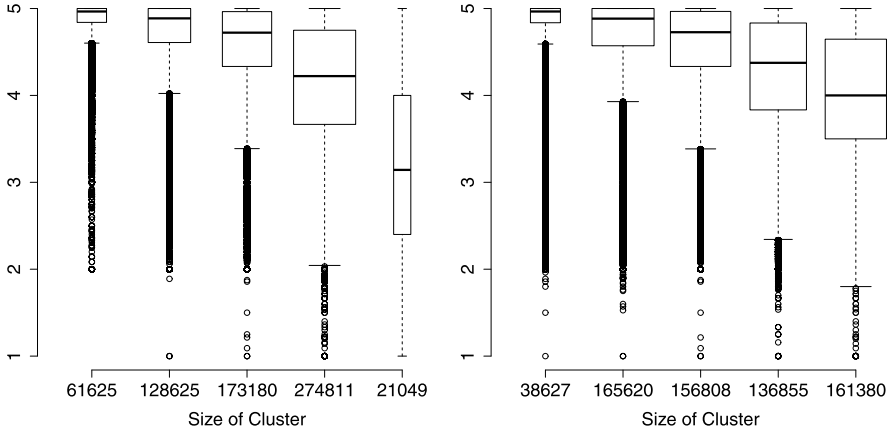
(b) Normal mixture model (7.3)



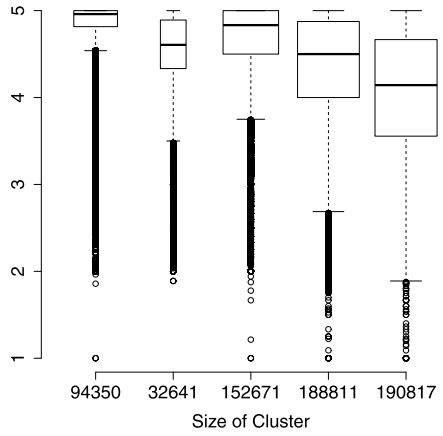
(c) Unconstrained network mixture model (2.9)

FIG. 3. *Observed values of excess trust* $e_i(\mathbf{y})$, *grouped by highest-probability component of* i, *for* (a) *parsimonious network mixture model* (7.2) *with* 12 *parameters*, (b) *normal mixture model* (7.3) *with* 14 *parameters*, *and* (c) *unconstrained network mixture model* (2.9) *with* 109 *parameters*.

As an objective criterion to compare the three models, we fit one-way ANOVA models where responses are article ratings and fixed effects are the group indicators of the articles' authors. The adjusted R^2 values are 0.262, 0.165 and 0.172 for the network mixture model, the normal mixture model and the unconstrained network mixture model, respectively. In other words, the latent structure detected by the 12-component network mixture model of (7.2) explains the variation in article ratings better than the 14-parameter univariate mixture model or the 109-parameter unconstrained network model.



(a) Network mixture model (7.2) (b) Normal mixture model (7.3)



(c) Unconstrained network mixture model (2.9)

FIG. 4. *Average ratings of* 659,290 *articles, grouped according to the highest-probability category of the article's author, for* (a) *parsimonious network mixture model* (7.2) *with* 12 *parameters*, (b) *normal mixture model* (7.3) *with* 14 *parameters, and* (c) *unconstrained network mixture model* (2.9) *with* 109 *parameters. The ordering of the five categories, which is the same as in Figure* 3*, indicates that the unconstrained network mixture model does not even preserve the correct ordering of the median average ratings.*

Table 1 reports estimates of the θ parameters from model (7.2) along with 95% confidence intervals reported in that table obtained by simulating 500 networks using the method of Section 5 and the parameter estimates obtained via our algorithm. For each network, we run our algorithm for 1000 iterations starting at the M-step, where the $\boldsymbol{\alpha}$ parameters are initialized to reflect the "true" component to which each node is assigned by the simulation algorithm by setting $\alpha_{ik} = 10^{-10}$ for k not equal to the true component and $\alpha_{ik} = 1 - 4 \times 10^{-10}$ otherwise. This is done to eliminate the so-called label-switching problem, which is rooted in

TABLE 1

95% Confidence intervals based on parametric bootstrap using 500 simulated networks, with 1000 iterations for each network. The statistic $\sum_i e_i(\mathbf{y})Z_{ik}$ equals $\sum_i \sum_{j\neq i} y_{ji}Z_{ik}$, where $Z_{ik} = 1$ if user i is a member of cluster k and $Z_{ik} = 0$ otherwise

Parameter	Statistic	Parameter estimate	Confidence interval
Negative edges (θ^-)	$\sum_{ij} y_{ij}^-$	-24.020	$(-24.029, -24.012)$
Positive edges (θ^+)	$\sum_{ij} y_{ij}^+$	0	—
Negative reciprocity (θ^{--})	$\sum_{ij} y_{ij}^- y_{ji}^-$	8.660	(8.614, 8.699)
Positive reciprocity (θ^{++})	$\sum_{ij} y_{ij}^+ y_{ji}^+$	9.899	(9.891, 9.907)
Cluster 1 trustworthiness (θ_1^Δ)	$\sum_i e_i(\mathbf{y})Z_{i1}$	-6.256	$(-6.260, -6.251)$
Cluster 2 trustworthiness (θ_2^Δ)	$\sum_i e_i(\mathbf{y})Z_{i2}$	-7.658	$(-7.662, -7.653)$
Cluster 3 trustworthiness (θ_3^Δ)	$\sum_i e_i(\mathbf{y})Z_{i3}$	-9.343	$(-9.348, -9.337)$
Cluster 4 trustworthiness (θ_4^Δ)	$\sum_i e_i(\mathbf{y})Z_{i4}$	-11.914	$(-11.919, -11.908)$
Cluster 5 trustworthiness (θ_5^Δ)	$\sum_i e_i(\mathbf{y})Z_{i5}$	-15.212	$(-15.225, -15.200)$

the invariance of the likelihood function to switching the labels of the 5 components and which can affect bootstrap samples in the same way it can affect Markov chain Monte Carlo samples from the posterior of finite mixture models [Stephens (2000)]. The sample 2.5% and 97.5% quantiles form the confidence intervals shown. In addition, we give density estimates of the five trustworthiness bootstrap samples in Figure 5. Table 1 shows that some clusters of users are much more trustworthy than others. In addition, there is statistically significant evidence

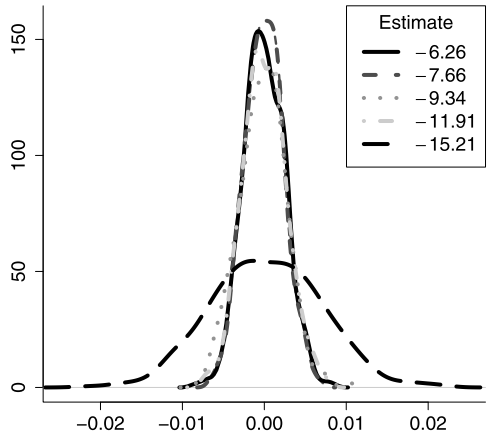


FIG. 5. *Kernel density estimates of the five bootstrap samples of the trustworthiness parameters, shifted so that each component's estimated parameter value (shown in the legend) equals zero.*

that users rate others in accordance with both *lex talionis* and *quid pro quo*, since both θ^{--} and θ^{++} are positive. These findings suggest that the ratings of pairs of users i and j are, perhaps unsurprisingly, dependent and not free of self-interest.

Finally, a few remarks concerning the parametric bootstrap are appropriate. While we are encouraged by the fact that bootstrapping is even feasible for problems of this size, there are aspects of our investigation that will need to be addressed with further research. First, the bootstrapping is so time-consuming that we were forced to rely on computing clusters with multiple computing nodes to generate a bootstrap sample in reasonable time. Future work could focus on more efficient bootstrapping. Some work on efficient bootstrapping was done by Kleiner et al. (2011), but it is restricted to simple models and not applicable here.

Second, when the variational GEM algorithm is initialized at random locations, it may converge to local maxima whose $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ values are inferior to the solutions attained when the algorithm is initialized at the "true" values used to simulate the networks. While it is not surprising that variational GEM algorithms converge to local maxima, it is surprising that the issue shows up in some of the simulated data sets but not in the observed data set. One possible explanation is that the structure of the observed data set is clear cut, but that the components of the estimated model are not sufficiently separated. Therefore, the estimated model may place nonnegligible probability mass on networks where two or more subsets of nodes are hard to distinguish and the variational GEM algorithm may be attracted to local maxima.

Third, some groups of confidence intervals, such as the first four trustworthiness parameter intervals, have more or less the same width. We do not have a fully satisfying explanation for this result; it may be a coincidence or it may have some deeper cause related to the difficulty of the computational problem.

In summary, we find that the clustering framework we introduce here provides useful results for a very large network. Most importantly, the sensible application of statistical modeling ideas, which reduces the unconstrained 109-parameter model to a constrained 12-parameter model, produces vastly superior results in terms of interpretability, numerical stability and predictive performance.

8. Discussion. The model-based clustering framework outlined here represents several advances. An attention to standard statistical modeling ideas relevant in the network context improves model parsimony and interpretability relative to fully unconstrained clustering models, while also suggesting a viable method for assessing precision of estimates obtained. Algorithmically, our advances allow us to apply a variational EM idea, recently applied to network clustering models in numerous publications [e.g., Airoldi et al. (2008), Daudin, Picard and Robin (2008), Mariadassou, Robin and Vacher (2010), Nowicki and Snijders (2001), Zanghi et al. (2010)], to networks far larger than any that have been considered to date. We have applied our methods to networks with over a hundred thousand nodes and signed edges, indicating how they extend to categorical-valued edges

generally or models that incorporate other covariate information. In practice, these methods could have myriad uses, from identifying high-density regions of large networks to selecting among competing models for a single network to testing specific network effects of scientific interest when clustering is present.

To achieve these advances, we have focused exclusively on models exhibiting dyadic independence conditional on the cluster memberships of nodes. It is important to remember that these models are *not* dyadic independence models overall, since the clustering itself introduces dependence. However, to more fully capture network effects such as transitivity, more complicated models may be needed, such as the latent space models of Hoff, Raftery and Handcock (2002), Schweinberger and Snijders (2003) or Handcock, Raftery and Tantrum (2007). A major drawback of latent space models is that they tend to be less scalable than the models considered here. An example is given by the variational Bayesian algorithm developed by Salter-Townshend and Murphy (2013) to estimate the latent space model of Handcock, Raftery and Tantrum (2007). The running time of the algorithm is $O(n^2)$ and it has therefore not been applied to networks with more than $n = 300$ nodes and $N = 89{,}700$ edge variables. An alternative to the variational Bayesian algorithm of Salter-Townshend and Murphy (2013) based on case-control sampling was proposed by Raftery et al. (2012). However, while the computing time of this alternative algorithm is $O(n)$, the suggested preprocessing step, which requires determining the shortest path length between pairs of nodes, is $O(n^2)$. As a result, the largest network Raftery et al. (2012) analyze is an undirected network with $n = 2716$ nodes and $N = 3{,}686{,}970$ edge variables.

In contrast, the running time of the variational GEM algorithm proposed here is $O(n)$ in the constrained and $O(f(n))$ in the unconstrained version of the Nowicki and Snijders (2001) model, where $f(n)$ is the number of edge variables whose value is not equal to the baseline value. It is worth noting that $f(n)$ is $O(n)$ in the case of sparse graphs and, therefore, the running time of the variational GEM algorithm is $O(n)$ in the case of sparse graphs. Indeed, even in the presence of the covariates, the running time of the variational GEM algorithm is $O(n \prod_{i=1}^{I} C_i)$ with categorical covariates, where I is the number of covariates and C_i is the number of categories of the ith covariate. We have demonstrated that the variational GEM algorithm can be applied to networks with more than $n = 131{,}000$ nodes and $N = 17$ billion edge variables.

While the running time of $O(n)$ shows that the variational GEM algorithm scales well with n, in practice, the "G" in "GEM" is an important contributor to the speed of the variational GEM algorithm: merely increasing the lower bound using an MM algorithm rather than actually maximizing it using a fixed-point algorithm along the lines of Daudin, Picard and Robin (2008) appears to save much computing time for large networks, though an exhaustive comparison of these two methods is a topic for further investigation.

An additional increase in speed might be gained by exploiting acceleration methods such as quasi-Newton methods [Press et al. (2002), Section 10.7], which

have shown promise in the case of MM algorithms [Hunter and Lange (2004)] and which might accelerate the MM algorithm in the E-step of the variational GEM algorithm. However, application of these methods is complicated in the current modeling framework because of the exceptionally large number of auxiliary parameters introduced by the variational augmentation.

We have neglected here the problem of selecting the number of clusters. Daudin, Picard and Robin (2008) propose making this selection based on the so-called ICL criterion, but it is not known how the ICL criterion behaves when the intractable incomplete-data log-likelihood function in the ICL criterion is replaced by a variational-method lower bound. In our experience, the magnitude of the changes in the maximum lower bound value achieved with multiple random starting parameters is at least as large as the magnitude of the penalization imposed on the log-likelihood by the ICL criterion. Thus, we have been unsuccessful in obtaining reliable ICL-based results for very large networks. More investigation of this question, and of the selection of the number of clusters in general, seems warranted.

By demonstrating that scientifically interesting clustering models can be applied to very large networks by extending the variational-method ideas developed for network data sets recently in the statistical literature, we hope to encourage further investigation of the possibilities of these and related clustering methods.

The source code, written in C++, and data files used in Sections 6 and 7 are publicly available at http://sites.stat.psu.edu/~dhunter/code.

APPENDIX A: OBTAINING A MINORIZER OF THE LOWER BOUND

The lower bound $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ of the log-likelihood function can be written as

$$\begin{aligned}\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}) = &\sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_{ik}\alpha_{jl} \log \pi_{d_{ij};kl}(\boldsymbol{\theta}) \\ &+ \sum_{i=1}^{n} \sum_{k=1}^{K} \alpha_{ik}(\log \gamma_k - \log \alpha_{ik}).\end{aligned} \tag{A.1}$$

Since $\log \pi_{d_{ij};kl}(\boldsymbol{\theta}) < 0$ for all $\boldsymbol{\theta}$, the arithmetic-geometric mean inequality implies that

$$\alpha_{ik}\alpha_{jl} \log \pi_{d_{ij};kl}(\boldsymbol{\theta}) \geq \left(\alpha_{ik}^2 \frac{\hat{\alpha}_{jl}}{2\hat{\alpha}_{ik}} + \alpha_{jl}^2 \frac{\hat{\alpha}_{ik}}{2\hat{\alpha}_{jl}}\right) \log \pi_{d_{ij};kl}(\boldsymbol{\theta}) \tag{A.2}$$

[Hunter and Lange (2004)], with equality if $\alpha_{ik} = \hat{\alpha}_{ik}$ and $\alpha_{jl} = \hat{\alpha}_{jl}$. In addition, the concavity of the logarithm function gives

$$-\log \alpha_{ik} \geq -\log \hat{\alpha}_{ik} - \frac{\alpha_{ik}}{\hat{\alpha}_{ik}} + 1 \tag{A.3}$$

with equality if $\alpha_{ik} = \hat{\alpha}_{ik}$. Therefore, function $Q_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{\alpha}; \hat{\boldsymbol{\alpha}})$ as defined in (3.8) possesses properties (3.9) and (3.10).

APPENDIX B: CONVEX DUALITY OF EXPONENTIAL FAMILIES

We show how closed-form expressions of $\boldsymbol{\theta}$ in terms of $\boldsymbol{\pi}$ can be obtained by exploiting the convex duality of exponential families. Let

$$\psi^*(\boldsymbol{\mu}) = \sup_{\boldsymbol{\theta}} \{\boldsymbol{\theta}^\top \boldsymbol{\mu} - \psi(\boldsymbol{\theta})\} \tag{B.1}$$

be the Legendre–Fenchel transform of $\psi(\boldsymbol{\theta})$, where $\boldsymbol{\mu} \equiv \boldsymbol{\mu}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}[\mathbf{g}(\mathbf{Y})]$ is the mean-value parameter vector and the subscripts k and l have been dropped. By Barndorff-Nielsen [(1978), page 140] and Wainwright and Jordan [(2008), pages 67 and 68], the Legendre–Fenchel transform of $\psi(\boldsymbol{\theta})$ is self-inverse and, thus, $\psi(\boldsymbol{\theta})$ can be written as

$$\psi(\boldsymbol{\theta}) = \sup_{\boldsymbol{\mu}} \{\boldsymbol{\theta}^\top \boldsymbol{\mu} - \psi^*(\boldsymbol{\mu})\} = \sup_{\boldsymbol{\pi}} \{\boldsymbol{\theta}^\top \boldsymbol{\mu}(\boldsymbol{\pi}) - \psi^*(\boldsymbol{\mu}(\boldsymbol{\pi}))\}, \tag{B.2}$$

where $\boldsymbol{\mu}(\boldsymbol{\pi}) = \sum_{d\in\mathcal{D}} \mathbf{g}(d)\pi_d$ and $\psi^*(\boldsymbol{\mu}(\boldsymbol{\pi})) = \sum_{d\in\mathcal{D}} \pi_d \log \pi_d$. Therefore, closed-form expressions of $\boldsymbol{\theta}$ in terms of $\boldsymbol{\pi}$ may be found by maximizing $\boldsymbol{\theta}^\top \boldsymbol{\mu}(\boldsymbol{\pi}) - \psi^*(\boldsymbol{\mu}(\boldsymbol{\pi}))$ with respect to $\boldsymbol{\pi}$.

APPENDIX C: GRADIENT AND HESSIAN OF LOWER BOUND

We are interested in the gradient and Hessian with respect to the parameter vector $\boldsymbol{\theta}$ of the lower bound in (A.1). The two examples of models considered in Section 2 assume that the conditional dyad probabilities $\pi_{d_{ij};kl}(\boldsymbol{\theta})$ take the form

$$\pi_{d_{ij};kl}(\boldsymbol{\theta}) = \exp\bigl[\boldsymbol{\eta}_{kl}(\boldsymbol{\theta})^\top \mathbf{g}(d_{ij}) - \psi_{kl}(\boldsymbol{\theta})\bigr], \tag{C.1}$$

where $\boldsymbol{\eta}_{kl}(\boldsymbol{\theta}) = \mathbf{A}_{kl}\boldsymbol{\theta}$ is a linear function of parameter vector $\boldsymbol{\theta}$ and $\mathbf{A}_{kl}$ is a matrix of suitable order depending on components k and l. It is convenient to absorb the matrix $\mathbf{A}_{kl}$ into the statistic vector $\mathbf{g}(d_{ij})$ and write

$$\pi_{d_{ij};kl}(\boldsymbol{\theta}) = \exp\bigl[\boldsymbol{\theta}^\top \mathbf{g}^\star_{kl}(d_{ij}) - \psi_{kl}(\boldsymbol{\theta})\bigr], \tag{C.2}$$

where $\mathbf{g}^\star_{kl}(d_{ij}) = \mathbf{A}_{kl}^\top \mathbf{g}(d_{ij})$. Thus, we may write

$$\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}) = \sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_{ik}\alpha_{jl}\bigl[\boldsymbol{\theta}^\top \mathbf{g}^\star_{kl}(d_{ij}) - \psi_{kl}(\boldsymbol{\theta})\bigr] + \text{const}, \tag{C.3}$$

where "const" denotes terms which do not depend on $\boldsymbol{\theta}$ and

$$\psi_{kl}(\boldsymbol{\theta}) = \log \sum_{d\in\mathcal{D}} \exp\bigl[\boldsymbol{\theta}^\top \mathbf{g}^\star_{kl}(d)\bigr]. \tag{C.4}$$

Since the lower bound $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ is a weighted sum of exponential family log-probabilities, it is straightforward to obtain the gradient and Hessian of $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ with respect to $\boldsymbol{\theta}$, which are given by

$$\nabla_\theta \mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}) = \sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_{ik}\alpha_{jl}\bigl\{\mathbf{g}^\star_{kl}(d_{ij}) - E_{\boldsymbol{\theta}}\bigl[\mathbf{g}^\star_{kl}(D_{ij})\bigr]\bigr\} \tag{C.5}$$

and

$$\nabla^2_\theta \mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha}) = -\sum_{i<j}^{n} \sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_{ik}\alpha_{jl} E_{\boldsymbol{\theta}}[\mathbf{g}^{\star}_{kl}(D_{ij})\mathbf{g}^{\star}_{kl}(D_{ij})^{\top}], \tag{C.6}$$

respectively.

In other words, the gradient and Hessian of $\mathrm{LB}_{\mathrm{ML}}(\boldsymbol{\gamma}, \boldsymbol{\theta}; \boldsymbol{\alpha})$ with respect to $\boldsymbol{\theta}$ are weighted sums of expectations—the means, variances and covariances of statistics. Since the sample space of dyads $\mathcal{D}$ is finite and, more often than not, small, these expectations may be computed by complete enumeration of all possible values of $d \in \mathcal{D}$ and their probabilities.

Acknowledgments. We are grateful to Paolo Massa and Kasper Souren of trustlet.org for sharing the epinion.com data

REFERENCES

ADAMIC, L. A. and GLANCE, N. (2005). The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery. LinkKDD'05* 36–43. ACM, New York.

AIROLDI, E., BLEI, D., FIENBERG, S. and XING, E. (2008). Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* **9** 1981–2014.

BARNDORFF-NIELSEN, O. (1978). *Information and Exponential Families in Statistical Theory*. Wiley, Chichester. MR0489333

BENAGLIA, T., CHAUVEAU, D., HUNTER, D. R. and YOUNG, D. (2009). mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software* **32** 1–29.

BESAG, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **36** 192–225.

BRITTON, T. and O'NEILL, P. D. (2002). Bayesian inference for stochastic epidemics in populations with random social structure. *Scand. J. Stat.* **29** 375–390. MR1925565

CAIMO, A. and FRIEL, N. (2011). Bayesian inference for exponential random graph models. *Social Networks* **33** 41–55.

CELISSE, A., DAUDIN, J.-J. and PIERRE, L. (2011). Consistency of maximum-likelihood and variational estimators in the stochastic block model. Preprint. Available at http://arxiv.org/pdf/1105.3288.pdf.

DAUDIN, J. J., PICARD, F. and ROBIN, S. (2008). A mixture model for random graphs. *Stat. Comput.* **18** 173–183. MR2390817

DAUDIN, J.-J., PIERRE, L. and VACHER, C. (2010). Model for heterogeneous random networks using continuous latent variables and an application to a tree-fungus network. *Biometrics* **66** 1043–1051. MR2758491

DAVIS, J. A. (1968). Statistical analysis of pair relationships: Symmetry, subjective consistency, and reciprocity. *Sociometry* **31** 102–119.

DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **39** 1–38. MR0501537

EFRON, B. (1979). Bootstrap methods: Another look at the jackknife. *Ann. Statist.* **7** 1–26. MR0515681

ERDŐS, P. and RÉNYI, A. (1959). On random graphs. I. *Publ. Math. Debrecen* **6** 290–297. MR0120167

FISHER, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **222** 309–368.

FRANK, O. and STRAUSS, D. (1986). Markov graphs. *J. Amer. Statist. Assoc.* **81** 832–842. MR0860518

GILBERT, E. N. (1959). Random graphs. *Ann. Math. Statist.* **30** 1141–1144. MR0108839

GROENDYKE, C., WELCH, D. and HUNTER, D. R. (2011). Bayesian inference for contact networks given epidemic data. *Scand. J. Stat.* **38** 600–616. MR2833849

HANDCOCK, M. (2003). Assessing degeneracy in statistical models of social networks. Technical report, Center for Statistics and the Social Sciences, Univ. Washington, Seattle. Available at http://www.csss.washington.edu/Papers.

HANDCOCK, M. S., RAFTERY, A. E. and TANTRUM, J. M. (2007). Model-based clustering for social networks. *J. Roy. Statist. Soc. Ser. A* **170** 301–354. MR2364300

HOFF, P. D., RAFTERY, A. E. and HANDCOCK, M. S. (2002). Latent space approaches to social network analysis. *J. Amer. Statist. Assoc.* **97** 1090–1098. MR1951262

HOLLAND, P. W. and LEINHARDT, S. (1981). An exponential family of probability distributions for directed graphs. *J. Amer. Statist. Assoc.* **76** 33–65. MR0608176

HUNTER, D. R. and HANDCOCK, M. S. (2006). Inference in curved exponential family models for networks. *J. Comput. Graph. Statist.* **15** 565–583. MR2291264

HUNTER, D. R. and LANGE, K. (2004). A tutorial on MM algorithms. *Amer. Statist.* **58** 30–37. MR2055509

KLEINER, A., TALWALKAR, A., SARKAR, P. and JORDAN, M. I. (2011). A scalable bootstrap for massive data. Preprint. Available at arXiv:1112.5016.

KOSKINEN, J. H., ROBINS, G. L. and PATTISON, P. E. (2010). Analysing exponential random graph (p-star) models with missing data using Bayesian data augmentation. *Stat. Methodol.* **7** 366–384. MR2643608

KUNEGIS, J., LOMMATZSCH, A. and BAUCKHAGE, C. (2009). The slashdot zoo: Mining a social network with negative edges. In *WWW'09: Proceedings of the 18th International Conference on World Wide Web* 741–750. ACM, New York.

MARIADASSOU, M., ROBIN, S. and VACHER, C. (2010). Uncovering latent structure in valued graphs: A variational approach. *Ann. Appl. Stat.* **4** 715–742. MR2758646

MASSA, P. and AVESANI, P. (2007). Trust metrics on controversial users: Balancing between tyranny of the majority and echo chambers. *International Journal on Semantic Web and Information Systems* **3** 39–64.

MCLACHLAN, G. and PEEL, D. (2000). *Finite Mixture Models*. Wiley, New York. MR1789474

MØLLER, J., PETTITT, A. N., REEVES, R. and BERTHELSEN, K. K. (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika* **93** 451–458. MR2278096

NEAL, R. M. and HINTON, G. E. (1993). A new view of the EM algorithm that justifies incremental and other variants. In *Learning in Graphical Models* 355–368. Kluwer Academic, Dordrecht.

NOWICKI, K. and SNIJDERS, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *J. Amer. Statist. Assoc.* **96** 1077–1087. MR1947255

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T. and FLANNERY, B. P. (2002). *Numerical Recipes in C++. The art of scientific computing*, 2nd ed. Cambridge Univ. Press, Cambridge. MR1880993

RAFTERY, A. E., NIU, X., HOFF, P. D. and YEUNG, K. Y. (2012). Fast inference for the latent space network model using a case-control approximate likelihood. *J. Comput. Graph. Statist.* **21** 901–919.

SALTER-TOWNSHEND, M. and MURPHY, T. B. (2013). Variational Bayesian inference for the latent position cluster model for network data. *Comput. Statist. Data Anal.* **57** 661–671. MR2981116

SCHWEINBERGER, M. (2011). Instability, sensitivity, and degeneracy of discrete exponential families. *J. Amer. Statist. Assoc.* **106** 1361–1370. MR2896841

SCHWEINBERGER, M., PETRESCU-PRAHOVA, M. and VU, D. Q. (2012). Disaster response on September 11, 2001 through the lens of statistical network analysis. Technical Report 116, Center for Statistics and the Social Sciences, Univ. Washington, Seattle.

SCHWEINBERGER, M. and SNIJDERS, T. A. B. (2003). Settings in social networks: A measurement model. In *Sociological Methodology* 33 (R. M. Stolzenberg, ed.) 307–341. Blackwell, Boston.

SNIJDERS, T. A. B. (2002). Markov chain Monte Carlo estimation of exponential random graph models. *Journal of Social Structure* **3** 1–40.

SNIJDERS, T. A. B. and NOWICKI, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *J. Classification* **14** 75–100. MR1449742

SNIJDERS, T. A. B., PATTISON, P. E., ROBINS, G. L. and HANDCOCK, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology* **36** 99–153.

STEFANOV, S. M. (2004). Convex quadratic minimization subject to a linear constraint and box constraints. *Appl. Math. Res. Express. AMRX* **1** 17–42. MR2064084

STEPHENS, M. (2000). Dealing with label switching in mixture models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **62** 795–809. MR1796293

STRAUSS, D. (1986). On a general class of models for interaction. *SIAM Rev.* **28** 513–527. MR0867682

STRAUSS, D. and IKEDA, M. (1990). Pseudolikelihood estimation for social networks. *J. Amer. Statist. Assoc.* **85** 204–212. MR1137368

TALLBERG, C. (2005). A Bayesian approach to modeling stochastic blockstructures with covariates. *J. Math. Sociol.* **29** 1–23.

WAINWRIGHT, M. J. and JORDAN, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* **1** 1–305.

WANG, B. and TITTERINGTON, D. M. (2005). Inadequacy of interval estimates corresponding to variational Bayesian approximations. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, *Jan* 6–8, 2005, *Savannah Hotel*, *Barbados* 373–380. Society for Artificial Intelligence and Statistics.

WASSERMAN, S. and PATTISON, P. (1996). Logit models and logistic regressions for social networks. I. An introduction to Markov graphs and p^*. *Psychometrika* **61** 401–425. MR1424909

ZANGHI, H., PICARD, F., MIELE, V. and AMBROISE, C. (2010). Strategies for online inference of model-based clustering in large and growing networks. *Ann. Appl. Stat.* **4** 687–714. MR2758645

D. Q. VU
DEPARTMENT OF MATHEMATICS
AND STATISTICS
UNIVERSITY OF MELBOURNE
PARKVILLE, VICTORIA 3010
AUSTRALIA
E-MAIL: duy.vu@unimelb.edu.au

D. R. HUNTER
DEPARTMENT OF STATISTICS
PENNSYLVANIA STATE UNIVERSITY
UNIVERSITY PARK, PENNSYLVANIA 16802
USA
E-MAIL: dhunter@stat.psu.edu

M. SCHWEINBERGER
DEPARTMENT OF STATISTICS
RICE UNIVERSITY
MS 138 P.O. BOX 1892
HOUSTON, TEXAS 77251-1892
USA