Structural Expectation Propagation (SEP): Bayesian structure learning for networks with latent variables

Nevena Lazic
Microsoft Research

Christopher M. Bishop
Microsoft Research,
Edinburgh University

John Winn
Microsoft Research

Abstract

Learning the structure of discrete Bayesian networks has been the subject of extensive research in machine learning, with most Bayesian approaches focusing on fully observed networks. One of the few methods that can handle networks with latent variables is the "structural EM algorithm" which interleaves greedy structure search with the estimation of latent variables and parameters, maintaining a single best network at each step.

We introduce Structural Expectation Propagation (SEP), an extension of EP which can infer the structure of Bayesian networks having latent variables and missing data. SEP performs variational inference in a joint model of structure, latent variables, and parameters, offering two advantages: (i) it accounts for uncertainty in structure and parameter values when making local distribution updates (ii) it returns a variational distribution over network structures rather than a single network, and . We demonstrate the performance of SEP both on synthetic problems and on real-world clinical data.

1 Introduction and Overview

A Bayesian network represents a multivariate distribution as a directed acyclic graph (DAG), such that the joint distribution factorizes into local distributions of single variables, conditioned on their parents in the graph. Bayesian networks are widely used tools in multivariate data analysis, in part because they allow conditional independencies and causal relationships between variables to be expressed in terms of graph properties. Learning the network structure from data has been the subject of extensive research, especially in the case of discrete networks whose local distributions are conditional probability tables. Approaches to this problem can broadly be grouped into constraint-based methods, which attempt to construct a network that satisfies a set of conditional independence constraints, and score-based methods, which aim to find the network that maximizes a penalized likelihood score. We use a Bayesian score-based approach and evaluate each network $\mathbf{G}$ according to its posterior probability given data, $\mathcal{D}$. This criterion is equivalent to the data marginal likelihood $p(\mathcal{D}|\mathbf{G})$ under a uniform prior over valid networks.

In fully observed networks, the marginal likelihood has the same factorization as the joint distribution and it can be computed in closed form, under standard independence and modularity assumptions on the model parameters [Heckerman et al., 1995]. This factorization has been exploited by many efficient structure learning algorithms, including greedy local searches [Heckerman et al., 1995, Chickering and Meek, 2002], dynamic programming [Koivisto, 2006, Tian et al., 2010], convex relaxations [Jaakkola et al., 2010], various sampling strategies [Friedman and Koller, 2000, Eaton and Murphy, 2007], and branch-and-bound algorithms [de Campos and Ji, 2011]. However, learning fully observed networks is mostly practical for problems of limited size. In complex high-dimensional systems, it is often useful or even essential to incorporate additional assumptions on structure regularities via parameter sharing and/or latent variables [Segal et al., 2003, Mansinghka et al., 2006, Shafto et al., 2011].

When the network includes latent variables, the marginal likelihood no longer decomposes and becomes intractable. Common approximations such as Cheeseman-Stutz [Cheeseman et al., 1988] and the variational Bayes lower bound [Beal and Ghahramani,

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31. Copyright 2013 by the authors.

2006] rely on a simpler variational distribution over latent variables and parameters. As this variational distribution needs to be re-inferred for every candidate network, efficient local search algorithms are no longer applicable. One possible strategy in this case is to alternate between optimizing structure given a fixed distribution over latent variables and parameters, and inferring latent variables and parameters for the current best structure. This is the idea behind the structural expectation maximization (SEM) algorithm and its variants [Friedman, 1998, Meila and Jordan, 2000, Thiesson, 1997, Elidan and Friedman, 2005]. A potential weakness of this approach is that it only considers a single structure at each iteration, making it susceptible to local optima. The more recent "cross-categorization model" [Shafto et al., 2011] uses sampling to obtain posterior distributions over structures. However, sampling can be slow, and the model is limited to bipartite networks in which each observed variable has a single latent parent.

In this paper we extend the highly successful Expectation Propagation (EP) [Minka, 2001] algorithm for inference over variables in a fixed network, to allow for joint inference over the structure of the network along with the latent variables and parameters. The posterior distribution over structure $\mathbf{G}$, latent variables $\mathbf{U}$, and parameters $\mathbf{\Theta}$ is estimated using a mean field variational distribution:

$$p(\mathbf{G}=\mathbf{g},\mathbf{U}=\mathbf{u},\mathbf{\Theta}=\theta|\mathcal{D}) \approxeq q(\mathbf{g})q(\mathbf{u})q(\theta). \qquad (1)$$

This approximation is still intractable, due to the exponential cardinality of $\mathbf{G}$. We reduce the size of the latent space by representing the network as a collection of discrete variables $\{G_1,...,G_D\}$, where each G_i indexes the parents of a variable X_i in the network. We approximate the network posterior by a factorized distribution over $G_1,...,G_D$:

$$q(\mathbf{g}) = \prod_i q(g_i). \qquad (2)$$

Although the cardinality of G_i can be exponential in general, this representation is manageable for many problems involving latent variable networks. For example, if we are interested in bipartite networks in which latent variables are the parents of observed variables and we limit the number of parents of each variable to two, $|G_i|$ is quadratic in the number of latent variables. This class of structures is sufficiently rich for many problems of interest.

In conventional Expectation Propagation, local distributions representing factors of a variational approximation to the true posterior distribution are updated iteratively. Each local update is informed by a 'context' given by the current variational factors over the rest of the network. In this paper, we extend this context to include uncertainty in network structure, where the structure is described using local discrete variables and represented using a graphical formalism called *gates* [Minka and Winn, 2008]. This allows us to extend EP to perform joint inference over structure, latent variables, and parameters. Each local inference is informed by a context comprising a variational posterior distribution over the remaining quantities. Thus, the distributions over latent variables and parameters are computed by probabilistically considering all possible networks, in contrast to structural EM which considers only a single network structure when updating variables and parameters.

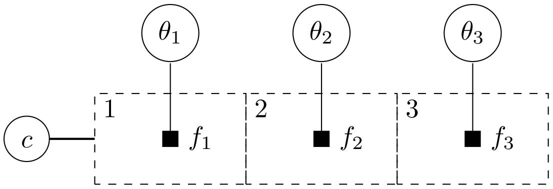

Figure 1: Gated factor graph for a mixture $g(c,\theta) = \prod_k f(\theta_k)^{\delta(c=k)}$. Only one of the factors f_k included in the model at a time, and the active factor is indexed by the variable c.

2 Inference Background

2.1 Gated Factor Graphs

To represent the model and derive inference updates, we use the graphical notation of gated factor graphs. Factor graphs [Kschischang et al., 2001] are bipartite graphs consisting of variable nodes and factor nodes. Each factor f_a evaluates a potential function over its neighbors $\mathbf{x}_a$ in the graph, and the joint distribution factorizes as the product of all potentials, $p(\mathbf{x}) = \prod_a f_a(\mathbf{x}_a)$. In some models, factor potentials contain additional structure that is not encoded in the graph; the simplest example is a mixture model:

$$g(c,\theta_1,...\theta_k) = \prod_k f_k(\theta_k)^{\delta(c=k)}. \qquad (3)$$

In a regular factor graph, g is connected to all mixture components θ_k, hiding the fact that only one is active at a time. Such context-specific independencies can be made explicit by augmenting the factor graph with the *gates* notation. A gate is a dashed rectangle that encloses a part of a factor graph and includes or excludes it from the model depending on the value of a selector variable, and a set of gates indexed by the same selector variable is a gate block. A gated factor graph for the mixture model of Equation 3 is shown in Figure 1.

2.2 Expectation Propagation

Expectation Propagation approximates a distribution $p(\mathbf{x}) = \prod_a f_a(\mathbf{x}_a)$ by a distribution with the same factorization $q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x}_a)$, where all factors are in the exponential family. EP iteratively refines the approximating factors through "deletion/inclusion" and moment matching steps, as follows. Each approximating factor $\tilde{f}_a(\mathbf{x}_a)$ is deleted to yield a partial posterior 'context' $q^{\setminus a}(\mathbf{x})$, and updated so that the following holds:

$$\tilde{f}_a(\mathbf{x}) q^{\setminus a}(\mathbf{x}_a) = \mathrm{proj}\big[f_a(\mathbf{x}) q^{\setminus a}(\mathbf{x}_a)\big]. \tag{4}$$

Here, the proj[$\cdot$] operator denotes the projection of its argument onto an exponential family distribution with matching moments.

When the approximation is fully factorized so that each approximating factor also factorizes as $\tilde{f}_a(\mathbf{x}) = \prod_i \tilde{f}_{ai}(x_i)$, each marginal $q(x_i)$ only depends on the factors that are functions of x_i, which we will denote by $ne(x_i)$. The contribution of each factor $f_a \in ne(x_i)$ can be thought of as a message $m_{ai}(x_i)$ received by x_i, and $q(x_i)$ is computed as the product of all such messages:

$$q(x_i) = \prod_{a, f_a \in ne(x_i)} m_{ai}(x_i). \tag{5}$$

In analogy to the sum-product algorithm [Kschischang et al., 2001], $m_a(x)$ corresponds to a factor-to-variable message, and $q^{\setminus a}(x)$ to a variable-to-factor message. EP iterative factor refinements correspond to the following message updates:

$$m_a(x_i) \propto \frac{\mathrm{proj}[\sum_{\mathbf{x}_a \setminus x_i} q^{\setminus a}(\mathbf{x}_a) f_a(\mathbf{x}_a)]}{q^{\setminus a}(x_i)} \tag{6}$$

In particular, a gate block factor of the form $g_a(c, \mathbf{x}_a) = \prod_k f_k(\mathbf{x}_a)^{\delta(c=k)}$ sends the following message to the selector variable c:

$$m_a(c = k) \quad \propto \quad \sum_{\mathbf{x}_a} f_k(\mathbf{x}_a) q^{\setminus a}(\mathbf{x}_a) \tag{7}$$

The message from $g_a(c, \mathbf{x}_a)$ to a variable $x_i \in \mathbf{x}_a$ has the following form:

$$m_a(x_i) \quad = \quad \frac{\mathrm{proj}[\sum_k q^{\setminus a}(c = k) r_k(x_i)]}{q^{\setminus a}(x_i)} \tag{8}$$

$$\text{where} \qquad r_k(x_i) = \sum_{\mathbf{x}_a \setminus x_i} f_k(\mathbf{x}_a) q^{\setminus a}(\mathbf{x}_a) \tag{9}$$

The EP approximation of the marginal likelihood is a product of contributions from all variables and factors. The contributions of a variable x_i, factor $f_a(\mathbf{x}_a)$, and gate block $g_b(c, \mathbf{x}_b)$ are respectively:

$$s_i \quad = \quad \sum_{x_i} q(x_i) \tag{10}$$

$$s_a \quad = \quad \frac{\sum_{\mathbf{x}_a} q^{\setminus a}(\mathbf{x}_a) f_a(\mathbf{x}_a)}{\sum_{\mathbf{x}_a} \prod_{x_j \in \mathbf{x}_a} q(x_j)} \tag{11}$$

$$s_b \quad = \quad \frac{\sum_k \sum_{\mathbf{x}_b} q^{\setminus b}(\mathbf{x}_b) f_k(\mathbf{x}_b)}{\sum_{\mathbf{x}_b} \prod_{x_j \in \mathbf{x}_b} q(x_j)} \tag{12}$$

3 Single-Parent Model

In this section, we fully specify the model of a bipartite discrete network in which latent variables are the parents of observed variables, and the structure is unknown. We assume that each observed variable is the child of a single latent parent, and defer the extension to multiple latent parents to Section 4.

Let $\mathbf{X} = \{X_1, ..., X_D\}$ and $\mathbf{U} = \{U_1, ..., U_K\}$ be the observed and latent variables, respectively. Let $G_i \in \{1, ..., K\}$ be a latent structure variable such that $G_i = k$ indicates that variable U_k is the parent of X_i, and let $\mathbf{G} = \{G_1, ..., G_D\}$. We assume uniform priors on the structure variables G_i, as well as on the latent variables U_k. Let Θ_{ij} be the multinomial parameters for the conditional probability of variable X_i given that its parent takes on the value j. We assume Dirichlet priors on Θ_{ij}, and use the shorthand notation $\Theta_i = \{\Theta_{i1}, ..., \Theta_{iJ}\}$ and $\Theta = \{\Theta_1, ..., \Theta_D\}$. We indicate the values taken on by random variables either using lowercase symbols, or by explicitly writing $X = x$.

Given N observations $\mathcal{D} = \{\mathbf{x}^1, ..., \mathbf{x}^N\}$, the joint distribution of structure, parameters, and variables can be written as:

$$p(\mathbf{g}, \theta, \mathbf{u}^{1:N}, \mathbf{x}^{1:N}) = \tag{13}$$

$$\prod_n \prod_i p(g_i) p(\theta_i) p(x_i^n) p(\mathbf{u}^n) h_{ni}(g_i, \mathbf{u}^n, \theta_i, x_i^n)$$

$$h_{ni}(g_i, \mathbf{u}^n, \theta_i, x_i^n) = \prod_k b_{nik}(u_k^n, \theta_i, x_i^n)^{\delta(g_i=k)} \tag{14}$$

$$b_{nik}(u_k^n, \theta_i, x_i^n) = \prod_j d_{nij}(\theta_{ij}, x_i^n)^{\delta(u_k^n=j)} \tag{15}$$

$$d_{nij}(\theta_{ij}, x_i^n) = \prod_l \theta_{ij,l}{}^{\delta(x_i^n=l)}. \tag{16}$$

Here, each factor $d_{nij}(\theta_{ij}, x_i^n)$ evaluates the probability of observation x_i^n given multinomial parameters θ_{ij}. Each factor $b_{nik}(u_k^n, \theta_i, x_i^n)$ is a discrete mixture model, selecting parameters θ_{ij} for x_i^n whenever the parent variable takes on the value j. Finally, each factor $h_{ni}(g_i, \mathbf{u}^n, \theta_i, x_i^n)$ selects the parent for X_i^n among variables $\{U_1^n, ..., U_K^n\}$ based on the coresponding structure variable G_i. The gated factor graph corresponding to this model is shown in Figure 2.

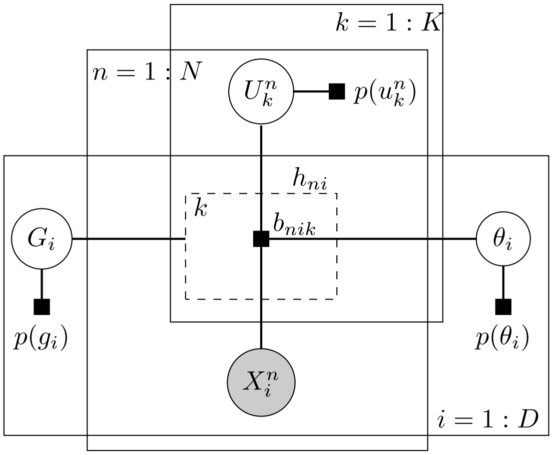

Figure 2: Gated factor graph representing a bipartite network in which each observed variable X_i has a single latent parent U_k, and the parent is indexed by G_i.

4 Inference

We use EP to estimate a factorized variational posterior distribution $q(\mathbf{g})q(\mathbf{u})q(\theta)$ over structure, latent variables, and parameters. We list the message updates here, and provide more detailed derivations in the supplementary material. As before, $q^{\setminus i}(x)$ denotes the posterior marginal of x after removing message from the factor indexed by i. In addition, we denote the expectation of a factor $f(x)$ under the distribution $q(x)$ by $E_{q(x)}[f(x)]$.

4.1 Messages to G_i

The posterior $q(G_i = k)$ over each structure variable is the product of the prior and the messages $\gamma_{ni}(G_i = k)$, $n = 1, ..., N$ from factors $h_{ni}(g_i, \mathbf{u}^n, \theta_i, x_i^n)$:

$$q(G_i = k) \propto p(G_i = k) \prod_n \gamma_{ni}(G_i = k) \qquad (17)$$

$$\gamma_{ni}(G_i = k) \propto \sum_j q^{\setminus i}(U_k^n = j) E_{q^{\setminus n}(\theta_{ij})}[d_{nij}(\theta_{ij}, x_i^n)]. \qquad (18)$$

Each message $\gamma_{ni}(G_i = k)$ is proportional to the evidence for a mixture model in which U_k^n is the parent of X_i^n, under a "leave-one-out" posterior over U_k^n and θ_i, computed using all observations except x_i^n. Expectations $E_{q(\theta)}[d(\theta, x)]$ can be computed in closed form; when $q(\theta)$ is parameterized by pseudocounts λ and λ_x is the pseudocount indexed by x, $E_{q(\theta)}[d(\theta, x)]$ evaluates to:

$$E_{q(\theta)}[d(\theta, x)] = \frac{\Gamma(\lambda_0)}{\Gamma(\lambda_0 + 1)} \frac{\Gamma(1 + \lambda_x)}{\Gamma(\lambda_x)} = \frac{\lambda_x}{\lambda_0}. \qquad (19)$$

4.2 Messages to U_k^n

The posterior of each latent variable $q(u_k^n)$ is the product of the prior and the messages $\nu_{nik}(u_k^n)$ from factors $h_{ni}(g_i, \mathbf{u}^n, \theta_i, x_i^n)$, $i = 1, ..., D$:

$$q(u_k^n) \propto p(u_k^n) \prod_i \nu_{nik}(u_k^n) \qquad (20)$$

$$\nu_{nik}(u_k^n) \propto \frac{\sum_{k'} q^{\setminus n}(G_i = k') r_{nik'}(u_k^n)}{q^{\setminus i}(u_k^n)}. \qquad (21)$$

Each message $\nu_{nik}(u_k^n)$ is a weighted average of the evidence for u_k^n given different structures, with weights given by the approximate posterior $q^{\setminus n}(G_i = k')$.

To compute the terms $r_{nik'}(u_k^n)$ following Eq. 9, we consider the cases $k' = k$ and $k' \neq k$ separately. When $k' \neq k$,

$$\begin{aligned} r_{nik'}(U_k^n = j) &\propto q^{\setminus i}(U_k^n = j) \qquad (22) \\ &\times \sum_{j'} q^{\setminus i}(U_{k'}^n = j') E_{q^{\setminus n}(\theta_{ij'})}[d_{nij'}(\theta_{ij'}, x_i^n)]. \end{aligned}$$

When $k' = k$,

$$r_{nik'}(U_k^n = j) = q^{\setminus i}(U_k^n = j) E_{q^{\setminus n}(\theta_{ij})}[d_{nij}(\theta_{ij}, x_i^n)]. \qquad (23)$$

4.3 Messages to θ_{ij}

Each parameter posterior distribution $q(\theta_{ij})$ is computed as the product of the prior and the messages $\rho_{nij}(\theta_{ij})$ from factors $h_{ni}(g_i, \mathbf{u}^n, \theta_i, x_i^n)$, $n = 1, ..., N$:

$$q(\theta_{ij}) = p(\theta_{ij}) \prod_n \rho_{nij}(\theta_{ij}) \qquad (24)$$

$$\rho_{nij}(\theta_{ij}) = \frac{\text{proj}[\sum_k q^{\setminus n}(G_i = k) s_{nijk}(\theta_{ij})]}{q^{\setminus n}(\theta_{ij})}. \qquad (25)$$

The message $\rho_{nij}(\theta_{ij})$ is a weighted average of Dirichlet messages, projected onto a Dirichlet distribution with matching moments (see [Minka, 2000] or [Minka and Lafferty, 2002] for details). The terms $s_{nijk}(\theta_{ij})$ are EP messages in a discrete mixture model where U_k is the parent of X_i. Each $s_{nijk}(\theta_{ij})$ is a moment-matched weighted average two Dirichlet distributions, for the two cases where $U_k^n = j$ and $U_k^n \neq j$:

$$\begin{aligned} s_{nijk}(\theta_{ij}) &= \text{proj}\big[q^{\setminus n}(\theta_{ij}) q^{\setminus i}(U_k^n = j) d_{nij}(\theta_{ij}, x_i^n) \\ &+ q^{\setminus n}(\theta_{ij}) \sum_{j' \neq j} q^{\setminus i}(U_k^n = j') E_{q^{\setminus n}(\theta_{ij'})}[d_{nij'}(\theta_{ij'}, x_i^n)]\big]. \end{aligned} \qquad (26)$$

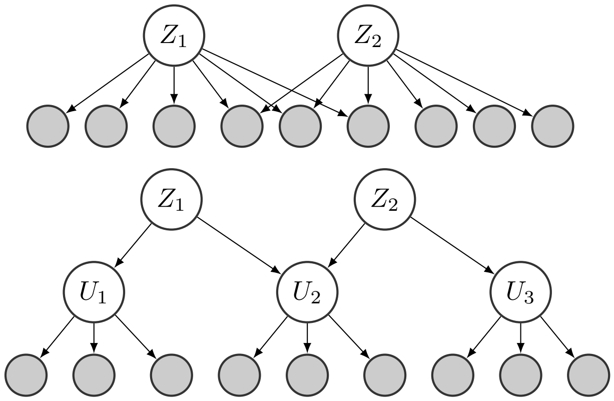

Figure 3: The top network can be converted to the bottom network via a one-to-one mapping from parent variables Z_1, Z_2 to child variables U_1, U_2, U_3.

4.4 Inference with Multiple Latent Parents

To allow observed variables to have multiple latent parents in the model, we only need to augment the inference algorithm with a few extra messages if we modify the network as follows. Let $\{Z_1, ..., Z_L\}$ be the set of latent variables in the model, and let each variable U_k deterministically encode the interaction of several variables $\mathbf{Z}_k$, as illustrated in Figure 3. A structure in which U_k is the parent of an observed variable X_i is then equivalent to a structure in which variables $\mathbf{Z}_k$ are the parents of X_i.

We encode the mapping from $\mathbf{Z}_k$ to U_k using indicator factors $I_{nk}(u_k^n, \mathbf{z}_{u_k}^n)$. The marginal of $Z_l^n \in \mathbf{Z}_k^n$ is computed from marginals of its children as follows:

$$q(z_l^n) = p(z_l^n) \prod_{k, Z_l \in \mathbf{Z}_k} \sum_{u_k^n, \mathbf{z}_k^n \setminus z_l^n} q^{\setminus I}(u_k^n) I_{nk}(u_k^n, \mathbf{z}_k^n), \tag{27}$$

where $q^{\setminus I}(u_k^n)$ is the posterior $q(u_k^n)$ computed using all messages except the one from $I_{nk}(u_k^n, \mathbf{z}_k^n)$. This extra message to U_k^n is

$$\mu(u_k^n) = \sum_{\mathbf{z}_{U_k}^n} \prod_{l, Z_l \in \mathbf{Z}_k} q^{\setminus k}(z_l^n) I_{nk}(u_k^n, \mathbf{z}_k^n), \tag{28}$$

where $q^{\setminus k}(z_l^n)$ is the marginal of variable Z_l^n computed using all of its children except U_k^n.

4.5 Handling Missing Values

Datasets corresponding to many real problems may include missing values. This is easy to handle in our framework: we can simply exclude messages from all factors $h_{ni}(g_i, \mathbf{u}^n, \theta_i, x_i^n)$ for which the observation x_i^n is missing.

5 Obtaining the MAP Network

A key feature of SEP is that the results are expressed as a posterior distribution over network structures. In some situations, however, we also seek the single most-probable structure, and the simplest approximation to this is to set each structure variable G_i to its mode. When some posterior structure marginals are multimodal, we can compare the marginal likelihood of networks corresponding to different modes and select the top ones. If this search space is too large, we can possibly reduce it using cutset conditioning, i.e. we can condition on one or more variables G_i taking on a particular value, re-infer the remaining variables, and repeat until we obtain a smaller set of solutions. Here we use a more subtle procedure to reduce the number of modes without making hard decisions on structure, which conditions on the latent variables U_k^n. Following inference, we set all latent variables U_k^n with confident unimodal marginals to their modes, and re-run inference over all other latent variables and parameters.

6 Experiments

We first evaluated SEP on synthetic data, generated by sampling networks in which observed variables had up to two latent parents. MAP networks found by SEP were compared to the solutions obtained by the SEM algorithm in terms of the data marginal likelihood and structural similarity to the true network. We also applied our approach to a real-world clinical dataset coming from an allergy study where patients were tested for allergic sensitization to a large number of different proteins, comprising components of common allergens. Here, structural inference helped discover subsets of proteins to which patients have similar allergic reactions as well as latent patient characteristics.

6.1 Synthetic Data

We generated synthetic data by sampling network structures and sampling data for each structure. Each network contained 50 observed variables, $K \in \{2, 3, 4, 5\}$ latent variables, and $N \in \{100, 200, 500, 1000\}$ datapoints. All latent variables were binary, all observed variables had cardinality four, and model parameters were set following [Chickering and Meek, 2002] to ensure variable dependence.[1] For each setting of K and N, we sampled 100 single-parent structures and 100 structures in which observed variables had up to two latent parents.

In initializing SEP, we constrained the maximum num-

[1] $P(U_k = 0) = 0.67$, $\theta_{i0} = [0.48, 0.24, 0.16, 0.12]$, and $\theta_{i1} = [0.12, 0.48, 0.24, 0.16]$

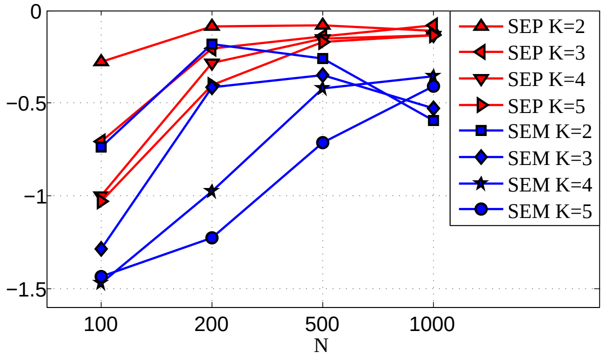

Figure 4: Average marginal likelihood log loss per data point for solutions obtained by SEP and SEM on datasets sampled from single-parent networks.

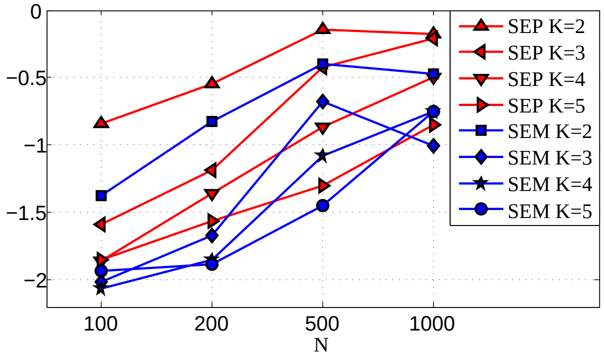

Figure 5: Average marginal likelihood log loss per data point for solutions obtained by SEP and SEM on datasets sampled from networks with up to two parents for each observed variable.

ber of parents of each observed variable to be either one (SEP1) or two (SEP2), as appropriate. For SEP1, we obtained a single structure simply by assigning each variable G_i to its MAP value. For SEP2, we applied the conditioning procedure described in Section 5: we set all latent variables with $q(U_k^n = j) > 0.9$ to their MAP values and re-inferred the remaining variables, repeated this one more time, and finally assigned structure variables to MAP values.

We compared our approach to an SEM implementation[2], where we used the Cheeseman-Stutz approximation of the marginal likelihood, greedy hill climbing network search, and a bipartite fully connected initial network. Constraining the number of parents was not possible for SEM as this would cause the greedy search to get stuck at initalization.

To evaluate the obtained solutions, we computed the average difference in the log marginal likelihood per data point between SEP/SEM solutions and the true network. Marginal likelihood was evaluated using the EP approximation for 1000 test data points sampled from the true network. The results are shown in Figures 4 and 5 as averages across 100 datasets for each setting of N and K. We also evaluated the structural similarity of the learned networks to the true network $\mathbf{g}^{\text{true}}$ for each dataset, in terms of the absolute difference $d(\mathbf{g}, \mathbf{g}^{\text{true}}) = \sum_{i,k} |g_{i,k} - g_{i,k}^{\text{true}}|$. These results are summarized in Tables 1 and 2, as averages across 100 datasets for each N and K, and qualitatively agree with the marginal likelihood score.

Table 1: Absolute difference in structure between the true network and SEP1 and SEM solutions for the single-parent datasets (mean across 100 networks).

N		K=2	K=3	K=4	K=5
100	SEP	**7.55**	**24.50**	**38.67**	**44.85**
	SEM	20.77	44.97	59.70	66.24
200	SEP	**2.34**	**4.54**	**11.95**	**18.32**
	SEM	5.97	13.97	40.72	57.46
500	SEP	**3.04**	**4.54**	**6.30**	**8.63**
	SEM	10.72	13.97	20.05	35.84
1000	SEP	**2.53**	**2.69**	**5.84**	**6.28**
	SEM	23.34	21.58	17.94	22.89

Table 2: Absolute difference in structure between the true network and SEP2 and SEM solutions for datasets with up to two parents for each observed variable (mean across 100 networks).

N		K=2	K=3	K=4	K=5
100	SEP	**25.01**	**53.83**	**70.16**	**80.76**
	SEM	36.40	64.08	77.29	84.72
200	SEP	**20.87**	**45.08**	**60.49**	**72.16**
	SEM	27.38	55.03	71.68	82.88
500	SEP	**17.02**	**32.79**	**53.00**	**67.33**
	SEM	18.99	36.5	60.12	75.55
1000	SEP	**13.3**	**34.74**	**53.56**	67.35
	SEM	21.15	43.43	58.01	**66.73**

SEP outperforms SEM in most settings, and the difference is greater for small datasets. We speculate that this is a consequence incorporating structural uncertainty in the estimation the latent variables and parameters, as well as only considering networks of lower complexity. Unsurprisingly, the performance of both methods deteriorates as the number of latent variables increases. In networks with two and three latent variables, SEM results become worse when the number of training datapoints is increased to 1000, suggesting possible overfitting.

For SEP, conditioning on a subset of latent variables and re-running inference generally led to improvements and disambiguation in structure whenever the initial

[2] available at http://compbio.cs.huji.ac.il/LibB/

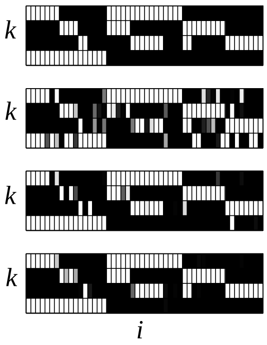

Figure 6: The effect of conditioning and re-running inference on two datasets with 4 latent variables and 500 data points. The top row shows the true structures, where a white rectangle at position (i, k) indicates the presence of an edge, i.e. $G_i = k$. The next three rows show the structure posteriors after running SEP2 with no conditioning, after one iteration of conditioning, and after two iterations of conditioning, top to bottom. Structural improvements typically occur if the initial solution sufficiently captures the ground truth.

solution was reasonably close to the true network. This is illustrated in Figure 6, where the first row shows two ground truth structures, the second row shows the structures inferred by SEP, and the next two rows show the posteriors after one and two iterations of conditioning. In the first network, the initial SEP solution is fairly similar to the ground truth and conditioning leads to structural improvements, while in the second network one of the latent variables is not adequately captured by the initial solution and conditioning cannot recover it.

SEP has a disadvantage when it comes to runtime and memory usage: both increase when we allow multiple latent parents, as a consequence of maintaining posterior distributions over all variables and their interactions. SEM is invariant to this, as it only maintains a single structure at each iteration. Although it is difficult to directly compare running times, we provide some reference values for $N = 1000$, $D = 50$, and $K = 3$. In this setting, the SEM compiled C++ executable took about 10 minutes for a dataset sampled from a network in which observed variables had one parent, and 16 minutes for a dataset in which observed variables had two parents. For the same datasets and on the same machine, our SEP1 and SEP2 implementations in C# relying on the Infer.NET library [Minka et al., 2010] took 10 minutes and 42 minutes respectively, with no conditioning re-runs.

Table 3: Log marginal likelihood of held-out clinical data given the learned structures (mean across 100 cross-validation splits).

K	SEM	SEP1	SEP2
2	-837.7	**-836.9**	-845.3
3	**-819.2**	-821.8	-828.0
4	-809.8	**-805.9**	-812.7
5	-804.8	-805.0	**-800.9**

6.2 Clinical Data

We used SEP to discover structure in clinical data obtained from a birth cohort study of asthma and allergies. In the study, $N = 221$ allergy-prone patients were tested for sensitization to $D = 71$ different allergen components. The testing was performed by measuring the response of IgE antibodies in blood to each component, and the results were categorized as negative, low, medium, or high, according to common clinical cutoffs.

We modeled this data by assuming a bipartite network in which binary latent variables are the parents of observed variables. Thus, learning network structure enabled us to discover groups of proteins to which patients have similar reactions, as well as to infer latent patient characteristics. To determine the number of latent variables K, we learned structure using SEM, SEP1, and SEP2 on 100 random subsets of 120 datapoints for $K \in \{2, 3, 4, 5\}$, and evaluated the solutions in terms of the marginal likelihood of the remaining 101 datapoints. For SEP, we used MAP solutions but treated all variables X_i such that $\max_k(G_i = k) < 0.5$ as independent. Based on the obtained results (see Table 3), we chose $K = 5$ for SEM and SEP2 and $K = 4$ for SEP1.

The networks inferred from all data are shown in Figure 7, along with the raw data sorted according to MAP values of the variables G_i and U_k^n in the SEP1 solution. An attractive feature of our solutions is that most of the posterior uncertainty corresponds to those allergen components to which there are few positive tests overall. SEP1 provides the simplest summary of the data, while SEM and SEP2 two solutions capture more of the feature subtleties, some of which may be an artefact of small sample size. The final network choice will depend on both the prior over structures and the marginal likelihood.

The inferred posterior distributions over structure and latent variables provide a useful way to visualize of complex high-dimensional clinical data, and can potentially lead to an improved understanding of patient characteristics. More information on this study can be obtained from the authors.

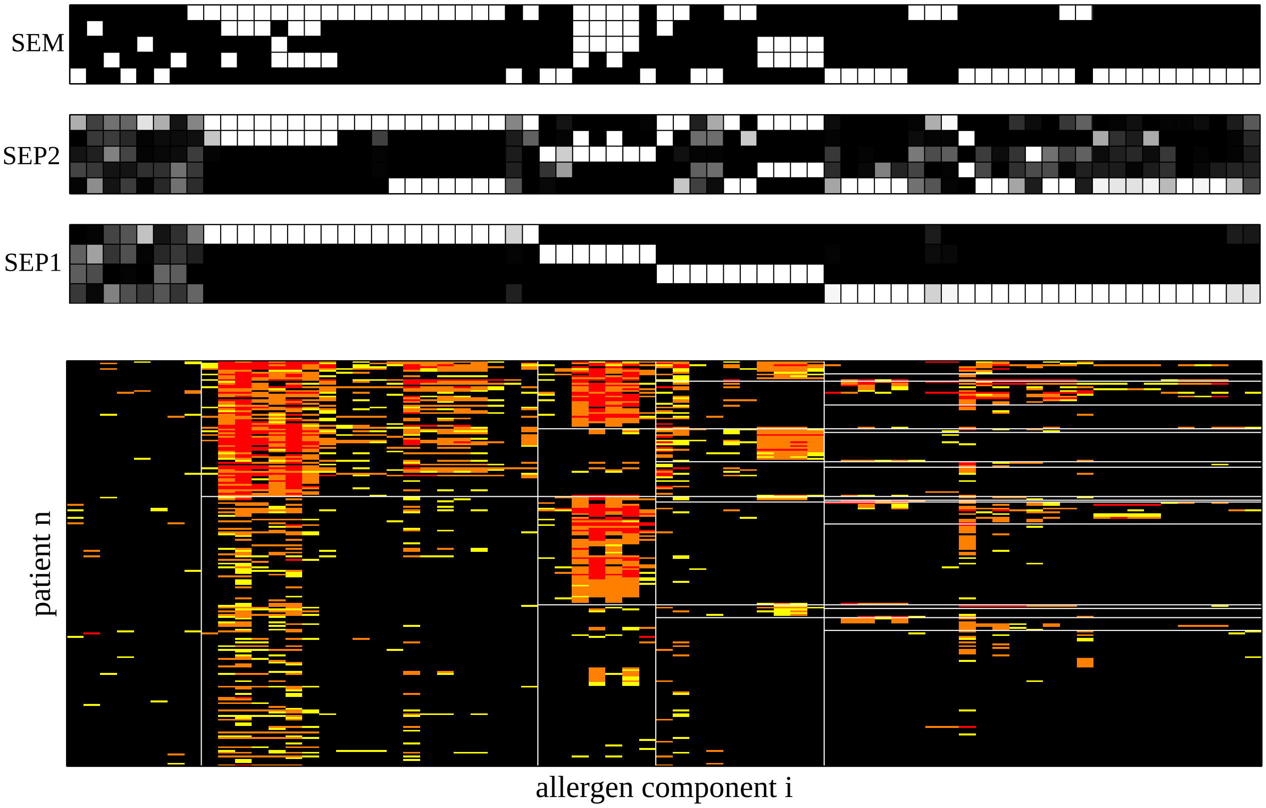

Figure 7: Top: networks learned by SEM and SEP on all clinical data, where a rectangle at position (i, k) indicates the posterior probability $q(G_i = k)$. Bottom: raw data with allergen components along the horizontal axis, patients along the vertical axis, and values color-coded as **negative**, low, medium, and high. Proteins are sorted according to MAP values of structure variables G_i, and patients according to MAP values of latent variables U_k^n in the SEP1 solution.

7 Discussion and Future Work

In this paper we have introduced Structural Expectation Propagation, a novel method for learning the structure of discrete latent variable networks. SEP iteratively updates a variational posterior distribution over networks, and this uncertainty in structure is taken into account in estimating latent variables and parameters. To the best of our knowledge, this is the first application of approximate Bayesian inference to this problem, and it demonstrably leads to improved inference of network structure. Although this comes at the expense of additional memory requirements and increased runtime, there exist strategies for improving computational efficiency. One possibility is to dynamically adapt the latent space during inference, for example by setting a subset of latent variables to their MAP values.

The bipartite networks we have considered in this paper correspond to a type of nonlinear dimensionality reduction for categorical variables. However, the framework can also be extended to hierarchical models, by allowing latent variables in the model to have their own latent parents and inferring the corresponding structure and parameters. Another potential direction for future work is to explore different structure priors to specify network constraints or preferences. For example, we can use the prior to constrain the total number of edges, to force a set of variables to share the same parents, or to assign variables to different parents. Beyond discrete networks, it would be interesting to apply Structural Expectation Propagation to problems involving different distributions, where some work already exists for sparse linear regression with spike-and-slab priors [Hernandez-Lobato et al., 2010].

Acknowledgements

The authors would like to thank the research team behind the Manchester Asthma and Allergy Study for providing the clinical data, as well as Tom Minka for technical comments.

References

[Beal and Ghahramani, 2006] Beal, M. J. and Ghahramani, Z. (2006). Variational bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis*, 1:793–831.

[Cheeseman et al., 1988] Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D. (1988). Autoclass: a bayesian classification system. In *Proc. Workshop on Machine Learning.*

[Chickering and Meek, 2002] Chickering, D. and Meek, C. (2002). Finding optimal bayesian networks. In *Proc. 18th Conference on Uncertainty in Artificial Intelligence (UAI).*

[de Campos and Ji, 2011] de Campos, C. and Ji, Q. (2011). Efficient structure learning of bayesian networks using constraints. *Journal of Machine Learning Research*, 12:663–689.

[Eaton and Murphy, 2007] Eaton, D. and Murphy, K. P. (2007). Bayesian structure learning using dynamic programming and mcmc. In *Proc. 23rd Conference on Uncertainty in Artificial Intelligence (UAI).*

[Elidan and Friedman, 2005] Elidan, G. and Friedman, N. (2005). Learning hidden variable networks: the information bottleneck approach. *Journal of Machine Learning Research*, 6:81–127.

[Friedman, 1998] Friedman, N. (1998). The bayesian structurall em algorithm. In *Proc. 14th Conference on Uncertainty in Artificial Intelligence (UAI).*

[Friedman and Koller, 2000] Friedman, N. and Koller, D. (2000). Being bayesian about network structure: a bayesian approach to structure discovery in bayesian networks. In *Proc. 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 201–210.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20:197–243.

[Hernandez-Lobato et al., 2010] Hernandez-Lobato, D., Hernandez-Lobato, J., and Suarez, A. (2010). Expectation propagation for microarray data classification. *Pattern Recognition Letters*, 31:1618–1626.

[Jaakkola et al., 2010] Jaakkola, T., Sontag, D., Globerson, A., and Meila, M. (2010). Learning bayesian network structure using lp relaxations. In *Proc. 13th International Conference on Artificial Intelligence and Statistics (AISTATS).*

[Koivisto, 2006] Koivisto, M. (2006). Advances in exact bayesian structure discovery in bayesian networks. In *Proc. 22nd Conference on Uncertainty in Artificial Intelligence (UAI).*

[Kschischang et al., 2001] Kschischang, F., Frey, B., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory 47(2)*, 47:498–519.

[Mansinghka et al., 2006] Mansinghka, V., Kemp, C., Tenenbaum, J., and Griffiths, T. (2006). Structured priors for structure learning. In *Proc. 22nd Conference on Uncertainty in Artificial Intelligence (UAI).*

[Meila and Jordan, 2000] Meila, M. and Jordan, M. I. (2000). Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48.

[Minka, 2000] Minka, T. (2000). Estimating a dirichlet distribution. Technical report, M.I.T.

[Minka, 2001] Minka, T. (2001). Expectation propagation for approximate inference. In *Proc. 17th Conference on Uncertainty in Artificial Intelligence (UAI).*

[Minka and Lafferty, 2002] Minka, T. and Lafferty, J. (2002). Expectation propagation for the generative aspect model. In *Proc. 18th Conference on Uncertainty in Artificial Intelligence (UAI).*

[Minka and Winn, 2008] Minka, T. and Winn, J. (2008). Gates: a graphical notation for mixture models. Technical report, Microsoft Research.

[Minka et al., 2010] Minka, T., Winn, J., Guiver, J., and Knowles, D. (2010). Infer.NET 2.4. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

[Segal et al., 2003] Segal, E., Shapira, M., Regev, A., Pe'er, D., Koller, D., and Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34:166–176.

[Shafto et al., 2011] Shafto, Kemp, Mansinghka, V., and Tenenbaum, J. (2011). A probabilistic model of cross-categorization. *Cognition*, 120:1–25.

[Thiesson, 1997] Thiesson, B. (1997). Score and information for recursive exponential models with incomplete data. In *Proc. 13th Conference on Uncertainty in Artificial Intelligence (UAI).*

[Tian et al., 2010] Tian, J., He, R., and Ram, L. (2010). Bayesian model averaging using the k-best bayesian network structures. In *Proc. 26th Conference on Uncertainty in Artificial Intelligence (UAI).*