Factorial Mixture of Gaussians and the Marginal Independence Model

Ricardo Silva
Department of Statistical Science
University College London
ricardo@stats.ucl.ac.uk

Zoubin Ghahramani
Department of Engineering
University of Cambridge
zoubin@eng.cam.ac.uk

Abstract

Marginal independence constraints play an important role in learning with graphical models. One way of parameterizing a model of marginal independencies is by building a latent variable model where two independent observed variables have no common latent source. In sparse domains, however, it might be advantageous to model the marginal observed distribution directly, without explicitly including latent variables in the model. There have been recent advances in Gaussian and binary models of marginal independence, but no models with non-linear dependencies between continuous variables has been proposed so far. In this paper, we describe how to generalize the Gaussian model of marginal independencies based on mixtures, and how to learn parameters. This requires a non-standard parameterization and raises difficult non-linear optimization issues.

1 CONTRIBUTION

We present a novel approach to learn multivariate distributions under marginal independence constraints. Such constraints result, for instance, from sparse latent variable models where observed variables are generated by a small combination of independent unobserved variables. The sparseness of the model implies that many pairs of observed variables $\{Y_i, Y_j\}$ will not be generated by any common hidden factor (e.g., as in the stroke model found by Wood et al. (2006)). That is, such pairs are marginally independent. In this paper, we are not interested in finding latent representations of the data, but in modeling the corresponding marginal distribution over the observed variables. Hence we will avoid representations with explicit latent variables. We will consider mixture of Gaussian distributions that have several marginal independence constraints. Our contribution is a parameterization that includes sparse factor analysis as a special case and algorithms for computing maximum likelihood and maximum a posteriori (MAP) estimators under this class. This provides flexible choices of models to complement the existing Gaussian (Drton and Richardson, 2003) and binary models (Drton and Richardson, 2008).

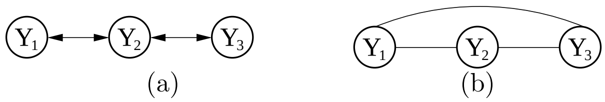

Figure 1: The graph in (a) indicates that variables Y_1 and Y_3 are marginally independent. The graph in (b) is the Markov network for the same distribution: since there are no conditional independencies encoded, the graph has to be complete.

Figure 1(a) illustrates a model of marginal independencies using a *bi-directed graph notation* (Drton and Richardson, 2003): the lack of a bi-directed edge $Y_i \leftrightarrow Y_j$ indicates that these two variables have to be marginally independent. Notice that models that factorize according to a connected Markov network cannot represent any marginal independence constraints (Figure 1(b)). The absence of an edge $Y_i - Y_j$ in an undirected network implies that Y_i and Y_j are conditionally independent given all other nodes, but marginally they will be dependent if the graph is connected. In contrast, the absence of an edge $Y_i \leftrightarrow Y_j$ in a bi-directed graph implies marginal independence of Y_i and Y_j, but in general these two variables will be conditionally dependent given all other variables.

The family of independence constraints encoded by the graphs in Huang and Frey (2008) happens to be the

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5. Copyright 2009 by the authors.

same as the one encoded by the bi-directed graph.

Unlike in models parameterized according to directed acyclic graphs (DAGs), there are no concerns about cycles in a bi-directed network. A marginal independence model is therefore complementary to other common graphical models. Some classes of models with *mixed* directed and bi-directed edges are a generalization of DAGs, with the advantage of being *closed under marginalization* (Richardson and Spirtes, 2002). For simplicity, in this paper we will consider pure bi-directed models only, as in Figure 1(a). For more applications and discussions of such models, see Richardson and Spirtes (2002); Bollen (1989); Silva and Ghahramani (2009); Huang and Frey (2008).

In Section 2, we discuss a parameterization of models of marginal independence. In Section 3, we describe a parameter learning algorithm. Experiments are described in Section 4. It is not the goal of this paper to present scalable algorithms. Off-the-shelf approximations such as basic mean field approaches cannot be applied directly in this case (see also, Silva and Ghahramani, 2009; Huang and Frey, 2008). Suitable approximations are out of the scope of the paper. Instead, we will focus on providing methods that work for small dimensional domains. Such methods will be useful in the future as a basis for approximations.

2 PARAMETRIC FORMULATION

Suppose we want to model a mixture of Gaussians where some variables are marginally independent, i.e., $f(\mathbf{Y}_i, \mathbf{Y}_j) = f(\mathbf{Y}_i)f(\mathbf{Y}_j)$ for two sets of marginally independent variables $\mathbf{Y}_i$ and $\mathbf{Y}_j$. Function $f(\mathbf{Y})$ is the respective marginal density of set $\mathbf{Y}$. We denote (conditional) independencies by Dawid's symbol $\perp\!\!\!\perp$. In this case, the independence is represented by $\mathbf{Y}_i \perp\!\!\!\perp \mathbf{Y}_j$.

The independence model for bi-directed graphs is described explicitly by Drton and Richardson (2008) and references within. Its *global Markov property* is defined as follows. Let $\mathcal{G}$ be a bi-directed graph with vertex set $\mathbf{Y}$. We say that set $\mathbf{Y}_i$ is independent of $\mathbf{Y}_j$ given $\mathbf{Y}_k$ if $\mathbf{Y}_i$ is separated from $\mathbf{Y}_j$ by $\mathbf{Y}\backslash(\mathbf{Y}_i \cup \mathbf{Y}_j \cup \mathbf{Y}_k)$. We denote independencies implied by the global Markov property on a bi-directed graph $\mathcal{G}$ by $\mathbf{Y}_i \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y}_j \mid \mathbf{Y}_k$.

One can also verify independence constraints over $\mathbf{Y}$ from the d-separation criterion (Richardson and Spirtes, 2002) applied to a directed acyclic graph (DAG) with vertex set $\mathbf{Y}$ augmented by (hidden) vertices $\mathbf{U}$. This DAG is created by replacing each bi-directed edge $Y_i \leftrightarrow Y_j$ by a directed path $Y_i \leftarrow U_{ij} \rightarrow Y_j$. For instance, the graph in Figure 1(a) can be reduced to the "canonical" DAG $Y_1 \leftarrow U_{12} \rightarrow Y_2 \leftarrow U_{23} \rightarrow Y_3$ before independencies are read-off. Notice that this relationship between bi-directed graphs and marginals of a DAG motivates the bi-directed edge representation.

In order to parameterize such a model, a first attempt would be to construct a sparse distribution for observed variables $\mathbf{Y}$ conditioned on some mixture indicator $c \in \{1, 2, \ldots, k\}$. The problem with this approach is made evident with the example in Figure 2(a) for $\mathbf{Y} = \{Y_1, Y_2, Y_3\}$: after integrating over the possible values of c, the constraint of interest, $Y_1 \perp\!\!\!\perp Y_3$, will be violated. This is illustrated in Figure 2(b).

2.1 The Factorial Model

A solution to the problem of preserving marginal independencies − while modeling non-linear, non-Gaussian distributions − is to build a *factorial mixture of Gaussians.* For each $Y_i \in \mathbf{Y}$, $i = 1, 2, \ldots, p$, define its *respective (discrete) indicator* c_i, with respective sample space $\{1, \ldots, k_i\}$. Let Π_c be the joint probability function of the mixture indicator variables, and let our model be given by

$$\begin{array}{rcl} \mathbf{c} & \sim & \Pi_c \\ \mathbf{Y} \mid \mathbf{c} & \sim & \mathcal{N}(\mu^{\mathbf{c}}, \Sigma^{\mathbf{c}}) \end{array} \quad (1)$$

where $\mathcal{N}(\mu, \Sigma)$ is the multivariate Gaussian distribution and the bold notation $\mathbf{c}$ denotes that we now have an indicator vector instead of an indicator scalar. Moreover, given a bi-directed graph $\mathcal{G}$ over $\mathbf{Y}$, the model (1) is subject to the following constraints[1]:

$$\begin{array}{rl} \mu_i^{\mathbf{c}} = \mu_i^{\mathbf{c}'}, & \text{if } c_i = c_i' \\ \sigma_{ij}^{\mathbf{c}} = \sigma_{ij}^{\mathbf{c}'}, & \text{if } c_i = c_i' \text{ and } \; c_j = c_j' \\ \sigma_{ij}^{\mathbf{c}} = 0, & \text{if edge } Y_i \leftrightarrow Y_j \text{ not in } \mathcal{G} \\ \mathbf{c}_i \perp\!\!\!\perp \mathbf{c}_j \mid \mathbf{c}_k, & \text{if } \mathbf{Y}_i \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y}_j \mid \mathbf{Y}_k \end{array} \quad (2)$$

The main result of this section can be summarized as follows:

Theorem 1 *Let $\mathcal{G}$ be a bi-directed graph and let a model $\mathcal{M}$ be given by (1) and (2), where $\mathcal{G}$ defines (2). If $\mathbf{Y}_i \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y}_j \mid \mathbf{Y}_k$, then $\mathcal{M}$ satisfies $\mathbf{Y}_i \perp\!\!\!\perp \mathbf{Y}_j \mid \mathbf{Y}_k$.*

Proof: We will show that $f(\mathbf{Y}_i, \mathbf{Y}_j, \mathbf{Y}_k) = g(\mathbf{Y}_i, \mathbf{Y}_k)h(\mathbf{Y}_j, \mathbf{Y}_k)$ for some $\{g(\cdot),\, h(\cdot)\}$, where $f(\cdot)$ is the density function of $\mathcal{M}$.

Assume $\mathbf{Y}_i \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y}_j \mid \mathbf{Y}_k$. By the definition of the global Markov property, no path $Y_i \leftrightarrow Y_{k0} \ldots Y_{k1} \leftrightarrow$

[1] Notice that in Gaussian models of marginal independence, the absence of an edge $Y_i \leftrightarrow Y_j$ in the bi-directed graph corresponds to the respective zero in the covariance matrix (i.e., $(\Sigma)_{ij} \equiv \sigma_{ij} = 0$). In the Markov network case, missing edges correspond to zeroes in the *inverse* covariance matrix (Drton and Richardson, 2003).

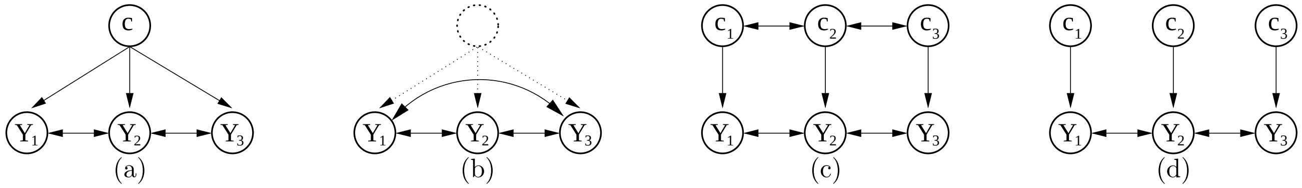

Figure 2: The mixture model according to (a), with c representing a mixture indicator, defines a sparse Gaussian distribution with independence $Y_1 \perp\!\!\!\perp Y_3$ on each mixture component. However, as illustrated in (b), marginalizing c will result in a model with no constraints, contrary to the intended assumptions. A solution is to adopt a *factorial* representation of mixture component membership. Here, the indicator of mixture membership is a vector $\mathbf{c} = \{c_1, c_2, c_3\}$ which will preserve the independence $Y_1 \perp\!\!\!\perp Y_3$ after marginalization. Adding further independencies to the distribution of $\mathbf{c}$ will preserve the original assumptions about $\mathbf{Y}$, as illustrated in (d).

Y_j can exist, where $Y_i \in \mathbf{Y}_i$, $Y_j \in \mathbf{Y}_j$ and $\{Y_{k0}, \dots, Y_{k1}\} \subseteq \mathbf{Y}_k$. This implies that we can partition $\mathbf{Y}_k$ into two sets $\mathbf{Y}_k^i$ and $\mathbf{Y}_k^j$, such that no vertex in $\mathbf{Y}_i \cup \mathbf{Y}_k^i$ is adjacent to any vertex in $\mathbf{Y}_j \cup \mathbf{Y}_k^j$.

Moreover, since by hypothesis each possible $\mathbf{Y}_i \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y}_j \mid \mathbf{Y}_k$ implies $\mathbf{c}_i \perp\!\!\!\perp \mathbf{c}_j \mid \mathbf{c}_k$, then $\Pi_{\mathbf{c}}$ is Markov with respect to a bi-directed graph $\mathcal{G}_{\mathbf{c}}$ with vertex set $\mathbf{c}$, where each edge $c_i \leftrightarrow c_j$ exists if and only if $Y_i \leftrightarrow Y_j$ is in $\mathcal{G}$. By definining $\mathbf{c}_i, \mathbf{c}_j, \mathbf{c}_k^i, \mathbf{c}_k^j$ analogously to the definitions in the previous paragraph, no vertex in $\mathbf{c}_i \cup \mathbf{c}_k^i$ is adjacent to any vertex in $\mathbf{c}_j \cup \mathbf{c}_k^j$, and consequently $\Pi(\mathbf{c}_i, \mathbf{c}_j, \mathbf{c}_k^i, \mathbf{c}_k^j)$ factorizes as $\Pi(\mathbf{c}_i, \mathbf{c}_k^i)\Pi(\mathbf{c}_j, \mathbf{c}_k^i)$.

Let $\Sigma_{ijk}^{\mathbf{c}}$ be the corresponding marginal covariance matrix for $\mathbf{Y}_i \cup \mathbf{Y}_j \cup \mathbf{Y}_k$ given $\mathbf{c}$. By the equality constraints in (2), $\Sigma_{ijk}^{\mathbf{c}}$ does not depend on $\mathbf{c} \backslash (\mathbf{c}_i \cup \mathbf{c}_j \cup \mathbf{c}_k)$. We denote this fact by representing this covariance matrix as $\Sigma_{ijk}^{\mathbf{c}_{ijk}}$. Similarly, we denote each $\mu_i^{\mathbf{c}}$ as $\mu_i^{c_i}$.

If $p_{\mathcal{N}}(\cdot)$ is the density function of a multivariate normal, the marginal density function $f(\mathbf{Y}_i, \mathbf{Y}_j, \mathbf{Y}_k)$ can then be written as:

$$
\begin{aligned}
&f(\mathbf{Y}_i, \mathbf{Y}_j, \mathbf{Y}_k) &= \\
&\sum_{\mathbf{c}_i \cup \mathbf{c}_j \cup \mathbf{c}_k^i \cup \mathbf{c}_k^j} p_{\mathcal{N}}(\mathbf{Y}_i, \mathbf{Y}_j, \mathbf{Y}_k^i, \mathbf{Y}_k^j \mid \mathbf{c}) \Pi(\mathbf{c}_i, \mathbf{c}_j, \mathbf{c}_k^i, \mathbf{c}_k^j) &= \\
&\left(\sum_{\mathbf{c}_i \cup \mathbf{c}_k^i} p_{\mathcal{N}}(\mathbf{Y}_i, \mathbf{Y}_k^i \mid \mathbf{c}_i, \mathbf{c}_k^i) \Pi(\mathbf{c}_i, \mathbf{c}_k^i) \right) &\times \\
&\left(\sum_{\mathbf{c}_j \cup \mathbf{c}_k^j} p_{\mathcal{N}}(\mathbf{Y}_j, \mathbf{Y}_k^j \mid \mathbf{c}_j, \mathbf{c}_k^j) \Pi(\mathbf{c}_j, \mathbf{c}_k^j) \right) &\equiv \\
&g(\mathbf{Y}_i, \mathbf{Y}_k) h(\mathbf{Y}_j, \mathbf{Y}_k)
\end{aligned}
$$

This follows from the fact that $\Sigma_{ijk}^{\mathbf{c}_{ijk}}$ is block-diagonal (with blocks indexed by $i \cup k^i$ and $j \cup k^j$) and that for arbitrary $\{Y_r, Y_s\}$ one can define the identity $\sigma_{rs}^{c_r c_s} \equiv \sigma_{rs}^{\mathbf{c}}$ (i.e., the respective covariance is indexed by $\{c_r, c_s\}$ only). □

As implied by the definition of the model, the parameterization of the distribution Π_c has also to obey the independence constraints in $\mathcal{G}$. One depiction of the joint model for $\mathbf{Y} \cup \mathbf{c}$ is shown in Figure 2(c) (for the precise semantics of models with directed and bi-directed edges, see Richardson and Spirtes, 2002). The parameterization described by Drton and Richardson (2008) could in principle be used to define Π_c. For simplicity, for the rest of the paper we assume that all mixture indicator variables are mutually independent, as depicted in Figure 2(d). Notice this does not violate the independencies required by $\mathcal{G}$.

2.2 Canonical Latent Variable Models

To illustrate how the given parameterization accounts for real-world phenomena, we briefly describe how a large class of latent variable DAG models is a special case of the bi-directed model.

Suppose that our observed dependencies all emerge from some DAG $\mathcal{G}_D$ with observed variables $\mathbf{Y}$ and latent variables $\{\mathbf{Z}, \mathbf{c}\}$. The model is given by

$$
Y_i = \lambda_{i0}^{c_i} + \sum_{Z_v \in parents(Y_i, \mathcal{G}_D) \backslash \mathbf{c}} \lambda_{iv}^{c_i} Z_v + \epsilon_i \qquad (3)
$$

where, as before, c_i is a discrete indicator. From a pool of possible coefficient parameters $\{\lambda\}$, $\mathbf{c} = \{c_1, c_2, \dots, c_p\}$ selects which parameters will be used when generating $\mathbf{Y} = \{Y_1, Y_2, \dots, Y_p\}$. Latent variables $\mathbf{Z}$ are multivariate Gaussian. Furthermore, ϵ_i is Gaussian distributed with zero mean and variance $v_i^{c_i}$ coming from a pool of possible variances also indexed by c_i. Without loss of generality, assume variables in $\mathbf{Z}$ have zero mean and unit variance. Elements in $\{\mathbf{Z}, \mathbf{c}\}$ are defined to be mutually independent.

For a latent variable DAG model specified this way, define a bi-directed graph $\mathcal{G}_{LV}$, with vertex set $\mathbf{Y}$, as

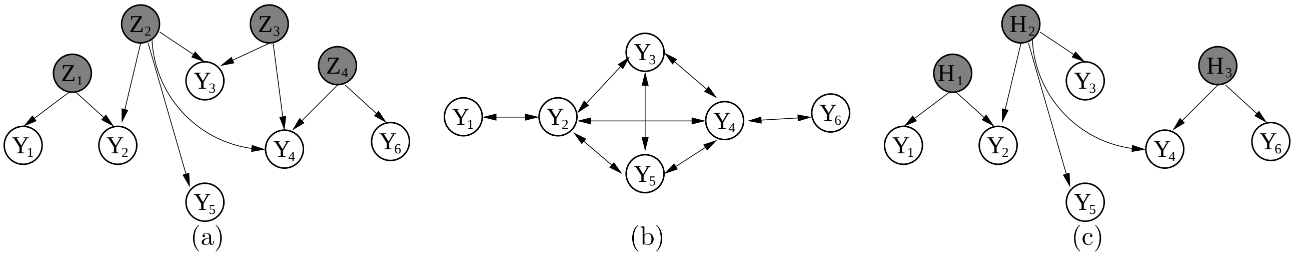

Figure 3: After marginalizing latent variables $\mathbf{Z}$ in (a), the result is the graph of marginal independencies shown in (b). If two variables belong to a same clique in (b), the graph accounts for the possibility of some hidden common parent from a DAG model that was marginalized. The encoding given by a bi-directed graph is agnostic with respect to the exact number of latent variables and how they connect to the elements of a clique. Given a bi-directed graph, there are different DAGs with latent variables corresponding to the same observable independencies. In (c), we illustrate a "canonical" DAG compatible with (b).

follows: add an edge $Y_i \leftrightarrow Y_j$ if and only if Y_i and Y_j have a common parent from $\mathbf{Z}$ in $\mathcal{G}_D$. Figures 3(a) and 3(b) illustrate this process.

From Equation (3), it follows that the covariance of Y_i and Y_j for a fixed choice of $\mathbf{c}$ is given by

$$\sigma_{ij}^{\mathbf{c}} = \sum_{Z_v \in \{parents(Y_i, \mathcal{G}_D) \cup parents(Y_j, \mathcal{G}_D)\} \backslash \mathbf{c}} \lambda_{iv}^{c_i} \lambda_{jv}^{c_j}$$

It is evident from this construction that $\sigma_{ij}^{\mathbf{c}}$ depends only on c_i and c_j. Similarly, it can be shown that the conditional mean and variance of Y_i, $\mu_i^{\mathbf{c}}$ and $\sigma_{ii}^{\mathbf{c}}$, depend only on c_i. Given that $\sigma_{ij}^{\mathbf{c}} = 0$ if $Y_i \leftrightarrow Y_j$ is not in $\mathcal{G}_{LV}$, and that c_i and c_j will be independent as well, the following result can be proved:

Proposition 1 *Any latent variable model defined by the above procedure can be parameterized by a factorial mixture of Gaussians model defined by the respective bi-directed graph $\mathcal{G}_{LV}$ and the conditions (1) and (2).*

It is not known if there is an exact latent variable model representation for every possible marginal independence model with factorial mixture of Gaussians. However, the parameterization based on (1) and (2) is agnostic with respect to any possible latent structure $\mathbf{Z}$ that generated the independence constraints. For comparison purposes, we define a *canonical* latent variable model from a bi-directed factorial mixture of Gaussians model as follows: for each clique in the bi-directed graph $\mathcal{G}$, create a latent variable Z in the DAG, and make it the parent of all vertices $\mathbf{Y}$ that lie in the clique. In Section 4, this will provide us with a gold standard to compare our factorial mixture of Gaussians to a default latent variable model that still respects the same marginal independence constraints. Figure 3 illustrates the different constructions.

3 THE LEARNING PROBLEM

Given the equality constraints in (2) and the assumption that Π_c fully factorizes, a bi-directed model defined by graph $\mathcal{G}$ with p vertices has the following *effective* parameters:

- the discrete marginal distribution $\pi_i(c_i)$ for each mixture component;
- the pool of mean parameters $\mu_i^{c_i}$ and variance parameters $\sigma_{ii}^{c_i}$, $1 \leq i \leq p$;
- the pool of covariance parameters $\sigma_{ij}^{c_i c_j}$, $1 \leq i < j \leq p$, such that $Y_i \leftrightarrow Y_j$ is in $\mathcal{G}$;

For simplicity, we assume that each c_i assumes values in same space $\{1, 2, ..., k\}$, where k is pre-specified. The number of parameters is then $O(|E|k^2 + pk)$, where $|E|$ is the number of edges in $\mathcal{G}$.

The learning problem is the maximum likelihood and MAP estimation of $\Theta = \{\{\pi(c_i)\}, \{\mu_i^{c_i}\}, \{\sigma_{ii}^{c_i}\}, \{\sigma_{ij}^{c_i c_j}\}\}$ given a training set $\mathcal{D} = \{\mathbf{Y}^{(1)}, \cdots, \mathbf{Y}^{(n)}\}$. Let $(\mathbf{Y}^{(d)}, \mathbf{c}^{(d)})$ denote a complete data point, $1 \leq d \leq n$ over the observables $\mathbf{Y}$ and indicators $\mathbf{c}$. Let $\langle \cdot \rangle_{f(\cdot)}$ be the expectation operator with respect to distribution $f(\cdot)$ and $\pi'(\mathbf{c}^{(d)})$ the conditional distribution of $\mathbf{c}^{(d)}$ given $\mathbf{Y}^{(d)}$ and a set of parameters Θ'.

In an expectation-maximization framework, we have to iteratively maximize the (penalized) likelihood given by Equation 6 in Table 3.1 with respect to Θ, subject to

$$\forall \mathbf{c}, \ \Sigma^{\mathbf{c}} \text{ is positive definite} \qquad (4)$$

where the (optional) introduction of penalization $\mathcal{P}(\Theta)$ corresponds to a prior distribution defined in the next section. The penalization is used if a maximum a posteriori estimator is required.

For the rest of the paper, let k be the maximum (prespecified) number of possible states among all c_i. The optimization problem stated above is $O(k^p)$. This problem is intractable in two ways. The first factor is the computation of the expectation in Equation 6 in Table 3.1 that requires a sum over a large number of terms. The second factor is the presence of an equally large number of constraints. Because the Markov blanket of any node in a connected bi-directed graph includes all other nodes (Silva and Ghahramani, 2009), the corresponding factor graph will include a factor adjacent to all nodes. Hence, off-the-shelf free-energy minimization approaches based on, e.g., belief propagation, are not directly applicable. This is an important area of future research, with some directions given by Huang and Frey (2008).

3.1 Priors

If a MAP estimator is required, a suitable choice of distribution for the prior of $\{\{\sigma_{ij}^{c_ic_j}\}, \{\sigma_{ii}^{c_i}\}\}$ is a product of experts (Hinton, 2002). We define a Gaussian expert for each non-zero covariance $\sigma_{ij}^{c_ic_j}$ and an inverse gamma expert for each variance $\sigma_{ii}^{c_i}$. The resulting prior is

$$\begin{array}{rcl} p(\{\sigma_{ij}\}, \{\sigma_{ii}\}) & \propto & \prod_{ij;c} p_{\mathcal{N}}(\sigma_{ij}^{c_ic_j}; m, v) \\ & \times & \prod_{i;c} p_{\mathcal{G}}(\sigma_{ii}^{c_i}; \alpha, \beta) \times \mathcal{I}(\{\sigma_{ij}\}, \{\sigma_{ii}\}) \end{array} \qquad (5)$$

where $p_{\mathcal{G}}(\cdot; \alpha, \beta)$ is an inverse gamma density function. The indicator function $\mathcal{I}(\cdot)$ returns zero if there is some $\Sigma^{\mathbf{c}}$ that is not positive definite. Our penalization term $\mathcal{P}(\Theta)$ is given by the logarithm of $p(\{\sigma_{ij}\}, \{\sigma_{ii}\})$. For simplicity, we are not considering priors for the mean parameters or mixture proportions.

3.2 Algorithms

We extend the maximum likelihood algorithm of Drton and Richardson (2003) for Gaussian distributions. The general idea is to perform iterative conditional fitting: fix the covariance matrix of a subset of variables $\mathbf{Y}_{\backslash i} \equiv \mathbf{Y} \backslash \{Y_i\}$ while optimizing for the row/column of Σ corresponding to Y_i. Drton and Richardson show that this local optimization can be done in closed form. Assume for now that the mean is zero. It is possible to define the conditional distribution of Y_i given $\mathbf{Y}_{\backslash i}$ by the regression

$$Y_i |\ \mathbf{Y}_{\backslash i} = \sum_{Y_j \ adjacent \ to \ Y_i} b_{ij} R_j + \zeta_i \qquad (8)$$

where R_j is the residual of the regression of Y_j on the nodes not adjacent to Y_i according to the current estimate of $\Sigma_{\backslash i}$, the covariance matrix of $\mathbf{Y}_{\backslash i}$. This implies that Σ_R, the covariance matrix of the set of residuals $\{R_j\}$, is given by a function of $\Sigma_{\backslash i}$. See (Drton and Richardson, 2003) for details.

By fixing $\Sigma_{\backslash i}$ (and, therefore, Σ_R), it is easy to update $\{b_{ij}\}$ and the variance of ζ_i by maximizing the conditional likelihood of Y_i. The corresponding i-th row and column of Σ can be then reconstructed from such parameters and $\Sigma_{\backslash i}$.

Conditional maximization in closed form is not possible anymore in the factorial model. The Markov blanket of Y_i includes all other variables (Silva and Ghahramani, 2009), and therefore for a fixed indicator vector $\mathbf{c}$, the equation corresponding to (8) becomes

$$Y_i \mid \{\mathbf{c},\ \mathbf{Y}_{\backslash i}\} = \sum_{Y_j \ adjacent \ to \ Y_i} b_{ij}^{\mathbf{c}} R_j^{\mathbf{c}} + \zeta_i^{\mathbf{c}} \qquad (9)$$

that is, the conditional parameters indexed by all entries of $\mathbf{c}$. However, there will be many equality constraints tying such parameters. For a given residual covariance matrix $\Sigma_R^{\mathbf{c}}$ and column vector of coefficients $b_i^{\mathbf{c}}$, we have

$$\sigma_{iv}^{\mathbf{c}} = \Sigma_{R,v}^{\mathbf{c}} b_i^{\mathbf{c}} \qquad \text{and} \qquad \sigma_{ii}^{\mathbf{c}} = \gamma_i^{\mathbf{c}} + b_i^{\mathbf{c}\mathsf{T}} \Sigma_R^{\mathbf{c}} b_i^{\mathbf{c}} \qquad (10)$$

where $\gamma_i^{\mathbf{c}}$ is the variance of $\zeta_i^{\mathbf{c}}$. Moreover, $\Sigma_{R,v}^{\mathbf{c}}$ is the row vector containing the respective covariances of each element of $\{R_j\}$ with Y_v according to $\Sigma^{\mathbf{c}}$.

The parameterization of the factorial mixture of Gaussians can be expressed as the constraints

$$\sigma_{ij}^{\mathbf{c}} = \sigma_{ij}^{\mathbf{c}'}, \text{if } c_i = c'_i \text{ and } c_j = c'_j \qquad (11)$$

$$\sigma_{ii}^{\mathbf{c}} = \sigma_{ii}^{\mathbf{c}'}, \text{if } c_i = c'_i \qquad (12)$$

These equalities define, respectively, linear and quadratic constraints on $\{b_i^{\mathbf{c}}\}$ for a fixed set $\{\gamma_i^{\mathbf{c}}\}$ (each $\gamma_i^{\mathbf{c}}$ has also to be positive). In particular, the equality of variances implies:

$$\gamma_i^{\mathbf{c}} + b_i^{\mathbf{c}T} \Sigma_R^{\mathbf{c}} b_i^{\mathbf{c}} = \gamma_i^{\mathbf{c}'} + b_i^{\mathbf{c}'T} \Sigma_R^{\mathbf{c}'} b_i^{\mathbf{c}'}, \text{if } c_i = c'_i \qquad (13)$$

Unlike in the Gaussian case, we cannot maximize function (9) for $\{b_i^{\mathbf{c}}\}$ separately from $\{\gamma_i^{\mathbf{c}}\}$ because of the constraints (12). For simplicity, we use the `fmincon` function in the MATLAB optimization library to perform constrained non-linear optimization. This is formalized as Algorithm 1 in Table 3.1.

So far, this assumed all mean parameters $\mu_i^{c_i}$ are zero. For the general case, the algorithm is essentially unmodified. We omit references to the mean parameters $\{\mu_i^{c_i}\}$ for simplicity[2].

[2] Within each instantiation of $\mathbf{c}^{(d)}$ in the expectation

Table 1: The general learning problem in each EM step is to maximize the objective function for Θ. Algorithm 1 is a generic template for solving this problem using the *conditional objective function*, where for simplicity we assume all mean parameters $\mu_i^{c_i}$ are zero. Matrices $\Sigma_{\backslash i}^{\mathbf{c}}$ are the marginal covariance matrices for $\mathbf{Y}_{\backslash i} \equiv \mathbf{Y} \backslash \{Y_i\}$. Penalization $\mathcal{P}(\Theta_{b,\gamma})$ corresponds to $\mathcal{P}(\Theta)$ by transforming variables.

Objective function: maximize

$$\mathcal{F}(\Theta; \mathcal{D}) = \sum_{d=1}^{n} \left\langle -\frac{1}{2} \log |\Sigma^{\mathbf{c}^{(d)}}| - \frac{1}{2} (\mathbf{Y}^{(d)} - \mu^{\mathbf{c}^{(d)}})^T \Sigma^{\mathbf{c}^{(d)}\,-1} (\mathbf{Y}^{(d)} - \mu^{\mathbf{c}^{(d)}}) + \log(\pi(\mathbf{c}^{(d)})) \right\rangle_{\pi'(\mathbf{c}^{(d)})} + \mathcal{P}(\Theta) \tag{6}$$

with respect to Θ, subject to: (4)

Conditional objective function: maximize

$$\mathcal{F}_i(\Theta_{b,\gamma}; \mathcal{D}) = \sum_{d=1}^{n} \left\langle -\frac{1}{2} \log(\gamma_i^{\mathbf{c}^{(d)}}) - \frac{1}{2\gamma_i^{\mathbf{c}^{(d)}}} \left(Y_i^{(d)} - \sum_{Y_j \text{ adjacent to } Y_i} b_{ij}^{\mathbf{c}^{(d)}} R_j^{\mathbf{c}^{(d)}} \right)^2 + \log(\pi(\mathbf{c}^{(d)})) \right\rangle_{\pi'(\mathbf{c}^{(d)})} + \mathcal{P}(\Theta_{b,\gamma}) \tag{7}$$

with respect to $\Theta_{b,\gamma}$, subject to: $\{\gamma_i > 0\}$, (11) and (12) (given (10))

Projection function: given current values for $\Theta_{b,\gamma} \equiv \{\{b_{ij}^{\mathbf{c}}\}, \{\gamma_i\}\}$ that respect constraints $\{\gamma_i^{\mathbf{c}} > 0\}$ and (11), but possibly not (12), change $\{\gamma_i^{\mathbf{c}}\}$ such that all three sets of constraints hold.

This is accomplished as follows. Select a subset $\mathcal{B}_\gamma \subseteq \{\gamma_i^{\mathbf{c}}\}$, of size k, to serve as a basis for $\{\gamma_i^{\mathbf{c}}\}$: we enumerate the values for $\mathbf{c}$ that correspond to the k-highest values of $b_i^{\mathbf{c}\,\mathsf{T}} \Sigma_R^{\mathbf{c}} b_i^{\mathbf{c}}$, and pick the corresponding $\gamma_i^{\mathbf{c}}$ to define $\mathcal{B}_\gamma$. The remaining elements of $\{\gamma_i^{\mathbf{c}}\}$ are updated by solving (12) with the fixed basis $\mathcal{B}_\gamma$ and fixed coefficients $\{b_{ij}^{\mathbf{c}}\}$. Return $\{\gamma_i^{\mathbf{c}}\}$.

Algorithm 1: takes data $\mathcal{D} = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(n)}\}$ and black-box constrained non-linear optimizer, FMINCON

1. Initialize Θ and compute $\pi'(\mathbf{c}^{(d)})$ for all $d = 1, 2, \ldots, n$
2. Iterate Steps 3 and 4 until convergence:
3. For $i = 1, 2, \ldots, p$: fix $\{\Sigma_{\backslash i}^{\mathbf{c}}\}$ and calculate $\{R_j^{\mathbf{c}}\}$, $\Theta_{b,\gamma} \equiv \{\{b_{ij}^{\mathbf{c}}\}, \{\gamma_i^{\mathbf{c}}\}\}$. Maximize $\mathcal{F}_i(\Theta_{b,\gamma}; \mathcal{D})$ with respect to $\Theta_{b,\gamma}$, subject to $\{\{\gamma_i^{\mathbf{c}} > 0\}, (11), (12)\}$ using FMINCON. Rebuild Θ from $\Sigma_{\backslash i}$ and $\Theta_{b,\gamma}$ using (10)
4. Maximize $\mathcal{F}(\Theta; \mathcal{D})$ with respect to $\{\pi_i(c_i)\}$ and set $\pi'(\mathbf{c}) \propto \mathcal{P}(\mathbf{Y} \mid \mathbf{c}) \prod_i \pi_i(c_i)$, the posterior of $\mathbf{c}$ given $\mathbf{Y}$.
5. Return Θ

Algorithm 2: same as **Algorithm 1**, except for Step 3

3. For $i = 1, 2, \ldots, p$: fix $\{\Sigma_{\backslash i}^{\mathbf{c}}\}$ and calculate $\{R_j^{\mathbf{c}}\}$, $\Theta_{b,\gamma} \equiv \{\{b_{ij}^{\mathbf{c}}\}, \{\gamma_i^{\mathbf{c}}\}\}$. Then
   i. Maximize $\mathcal{F}_i(\Theta_{b,\gamma}; \mathcal{D})$ with respect to $\{b_{ij}^{\mathbf{c}}\}$ subject to (11) by closed form/FMINCON
   ii. Initialize $\{\gamma_i^{\mathbf{c}}\}$ from the solution for $\{b_{ij}^{\mathbf{c}}\}$ by using the **Projection** function, and maximize $\mathcal{F}_i(\Theta_{b,\gamma}; \mathcal{D})$ with respect to $\{\gamma_i^{\mathbf{c}}\}$ subject to $\{\gamma_i^{\mathbf{c}} > 0\}$ and (12) using FMINCON

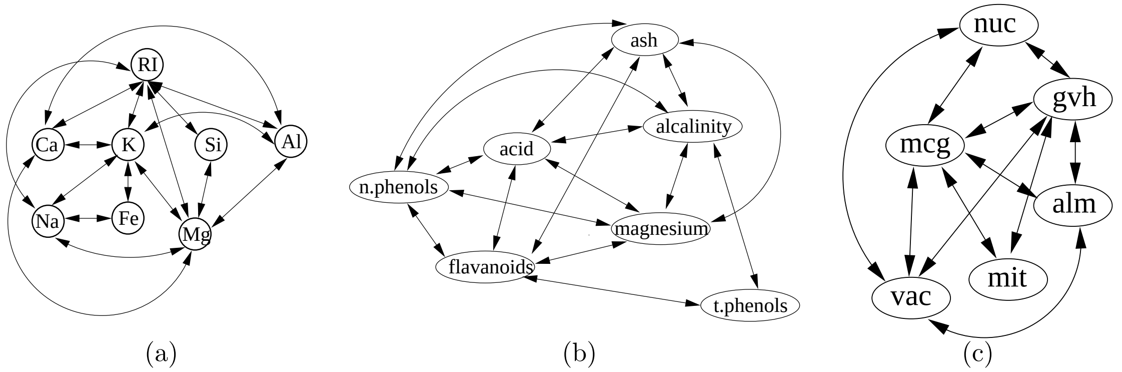

Figure 4: Typical networks found by non-parametric tests of marginal independencies for the GLASS, WINE and YEAST domains, respectively.

Algorithm 2 in Table 3.1 is a suggested relaxed variation of the problem. We maximize the conditional density of Y_i by first dropping the constraints (12) and fixing $\{\gamma_i^{\mathbf{c}}\}$. In maximum likelihood problems, where $\mathcal{P}(\Theta) \equiv 0$, this is a quadratic program with linear constraints (11) and can be solved analytically. To optimize for $\{\gamma_i^{\mathbf{c}}\}$ after fixing $\{b_{ij}^{\mathbf{c}}\}$, we need to project the current parameter vector $\Theta_{b,\gamma}$ back to the feasible space. This is a matter of enforcing (12) and the positivity of the variances $\{\gamma_i^{\mathbf{c}}\}$. One can verify that the function Projection in Table 3.1 accomplishes that. There is no guarantee this will increase the expected log-likelihood. In practice, if after the projection we decrease the conditional log-likelihood we had before optimizing for $\{b_{ij}^{\mathbf{c}}\}$, we reset the values of such parameters and use the complete optimization procedure of Step 3 of Algorithm 1 for the particular i. Although it seems in principle that we are taking the risk of unnecessarily paying for an overhead, this relaxed optimization often increases the log-likelihood and it is much less computationally expensive for large sample sizes (since for the MLE it requires only one pass through the data to find $\{b_{ij}^{\mathbf{c}}\}$) compared against the full step. By watching the steps where the optimization fails and changing the method accordingly, we are guaranteed to converge: only partial maximization at each step is necessary for the expectation-maximization method.

Sequential plug-in estimator: We also experiment with a simplified estimation criterion that exploits the fact that our model is defined by "upward compatible" parameters. That is, unlike standard undirected models, where parameters are not locally encoding small subsets of the marginals, our model can exploit the fact that *each parameter is encoding only a local joint.* Given an ordering $\{Y_1, Y_2, \dots, Y_p\}$ and an estimate for the parameters of the marginal $\{Y_1, Y_2, \dots, Y_{t-1}\}$, we apply Algorithm 1 for $i = t$ only. The process is implemented by varying t from 1 to p. This is the equivalent of carrying plug-in estimates for the marginal of $\{Y_1, \dots, Y_{t-1}\}$ into the learning of the parameters $\{\sigma_{tj}^{c_t c_j}, j < t\}, \{\mu_t^{c_t}\}, \pi(c_t)$. The motivation is to trade-off statistical efficiency for computational efficiency without introducing an approximation bias. The choice of an ordering is left open. In one of our experiments, we treat as Y_1 the variable with the fewest number of neighbors in the graph, Y_2 as the one with the second smallest neighborhood size, and so on. Notice that sequential optimization is not applicable in latent variable models, since in this case a same parameter has different roles in different marginals.

4 EXPERIMENTS

We first compare our model against the standard mixture of Gaussians (MG) and a latent variable model (LVM) fit by expectation-maximization (EM) using predictive log-likelihood as the criterion. Four UCI Machine Learning repository datasets (Asuncion and Newman, 2007) were chosen: GLASS ("Glass Identification"), FIRE ("Forest Fires"), HEART ("Statlog Heart") and WINE. The datasets were chosen so that they were small and had mostly continuous variables. We show that using the bi-directed graph parameterization can help to find better parameters for prediction, even after being initialized by the parameters of a latent variable model. Training is done by maximum likelihood. Mixture level k is set to 2, with efficient model selection left for future work.

A 5-fold cross validation procedure is performed. In order to provide a bi-directed graph for a given training set, we start with a complete graph and run the marginal independence test of Gretton et al. (2007) for each pair of variables. For each pair where the null

$\langle \cdot \rangle_{\pi'(\mathbf{c}(d))}$, we have first to temporarily subtract the mean $\mu^{\mathbf{c}(d)}$ from the data point $\mathbf{Y}^{(d)}$ before optimization for the covariances. After adding the means back, optimizing for $\{\mu_i^{c_i}\}$ is a standard unconstrained procedure and omitted from the discussion.

Table 2: Comparison of predictive log-likelihood of the bi-directed model (BDM) against the respective canonical latent variable model (LVM) and a mixture of Gaussians (MG). Each row corresponds to a test in a 5-fold cross-validation setup. The log-likelihood is averaged over the number of test points. Best results in bold.

Fold	Glass			Fire			Heart			Wine		
	BDM	LVM	MG	BDM	LVM	MG	BDM	LVM	MG	BDM	LVM	MG
1	**-4.73**	-5.61	-5.73	**-8.75**	-8.97	-9.62	**-2.59**	-3.64	-3.50	-8.50	**-8.46**	-8.71
2	**-0.09**	-1.56	-8.67	-7.11	-7.50	**-7.03**	**-4.88**	-5.05	-5.98	-8.45	**-8.44**	-8.52
3	**-0.04**	-2.06	-4.22	**-6.35**	-7.05	-6.63	**-3.00**	-3.18	-5.14	-8.55	**-8.51**	-8.67
4	**-3.34**	-4.59	-5.01	**-7.69**	-8.22	-7.88	-2.93	-3.84	**-2.67**	-10.0	-10.0	**-9.70**
5	**5.50**	4.44	-0.65	-7.44	-7.82	**-7.03**	**-0.09**	-1.49	-3.59	-8.23	-8.35	**-7.81**

hypothesis of independence is not rejected at a 0.05 level, we remove the respective edge. Typical graphs are shown in Figure 4. The latent variable model is the canonical one defined in Section 2. We train the bi-directed model by starting from the means and covariance matrices implied by the latent variable model: hence, our initial training log-likelihood is always the same final log-likelihood of the latent variable model. Results are summarized in Table 2. Algorithm 2 is used: the relaxation step increases the log-likelihood around half of the time (as discussed, we retry with the corresponding step in Algorithm 1 otherwise). As seen by the wine experiment (the smallest dataset, with 143 training points only), sometimes our model can overfit the data compared to LVM. Improvements over both methods are steady, and overall, the bi-directed model has a solid performance compared to LVM/MG.

Maximum a posteriori experiment: We used the yeast dataset from the UCI Machine Learning Repository. The yeast dataset contains 1484 datapoints and 7 continuous variables. We excluded the attribute "pox" for being independent of all other attributes according to the test of Gretton et al. (2007) at a 0.05 level. In our setup, we used two mixture components per node and five-fold cross-validation. Structure learning was done as described before. We compare results in MAP estimation against a latent variable model generated from bi-directed graphs in the same way as in the previous section. Since the dataset is comparatively larger, it is now safer to use the sequential optimization algorithm. The prior consisted on a product of (2, 2) inverse gammas and standard Gaussians. The resulting predictive log-likelihood for the five folds were -7.18, -7.15, -7.09, -7.24, -7.31 for the bi-directed model. For the LVM trained by maximum likelihood and EM, the results were -10.78, -10.24, -9.68, -10.04, -10.28, showing a sizeable difference.

5 CONCLUSION

This paper follows the spirit of Drton and Richardson (2008), where an exact algorithm for binary models of marginal independencies is derived. In both cases, the problem scales at an exponential rate with the number of variables. However, deriving an exact algorithm is an important step in order to design future approximation techniques, which might require non-standard approaches. Since the model is parameterized by marginal parameters instead of conditional ones, methods based on fitting sub-marginals and propagating estimates might be promising. We emphasize that the class of models here proposed are aimed at fairly sparse domains, and should be seen as a complement to latent variable models. Perhaps the most important application of bi-directed models is as a component of mixed graph formulations (Silva and Ghahramani, 2009). In this case, even the exact algorithm might be useful in a high dimensional problem: the bi-directed graphical component is often decomposable into small disconnected subgraphs. We plan to embed the techniques introduced here in mixed graph problems, for both MAP and fully Bayesian learning.

References

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

K. Bollen. *Structural Equation Models with Latent Variables*. John Wiley & Sons, 1989.

M. Drton and T. Richardson. A new algorithm for MLE in Gaussian models for marginal independence. *UAI*, 2003.

M. Drton and T. Richardson. Binary models for marginal independence. *J. of the Royal Stat. Society B*, 2008.

A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel stat. test of ind. *NIPS*, 2007.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

J. C. Huang and B. Frey. Cumulative distribution networks and the derivative-sum-product algorithm. *UAI*, 2008.

T. Richardson and P. Spirtes. Ancestral graph Markov models. *Annals of Statistics*, 30:962–1030, 2002.

R. Silva and Z. Ghahramani. The hidden life of latent variables: Bayesian learning with mixed graph models. *Journal of Machine Learning Research, to appear*, 2009.

F. Wood, T. Griffiths, and Z. Ghahramani. A non-param. Bayes. method for inferring hidden causes. *UAI*, 2006.