Computationally Efficient Methods for Selecting Among Mixtures of Graphical Models

B. THIESSON, C. MEEK, D. M. CHICKERING, and D. HECKERMAN
Microsoft Research, USA

Abstract

We describe computationally efficient methods for Bayesian model selection. The methods select among mixtures in which each component is a directed acyclic graphical model (mixtures of DAGs or MDAGs), and can be applied to data sets in which some of the random variables are not always observed. The model-selection criterion that we consider is the posterior probability of the model (structure) given data. Our model-selection problem is difficult because (1) the number of possible model structures grows super-exponentially with the number of random variables and (2) missing data necessitates the use of computationally slow approximations of model posterior probability. We argue that simple search-and-score algorithms are infeasible for a variety of problems, and introduce a feasible approach in which parameter and structure search is interleaved and expected data is treated as real data. Our approach can be viewed as the combination of (1) a modified Cheeseman–Stutz asymptotic approximation for model posterior probability and (2) the Expectation–Maximization algorithm. We evaluate our procedure for selecting among MDAGs on synthetic and real examples.

Keywords : Model selection, asymptotic methods, mixture models, directed acyclic graphs, hidden variables, EM algorithm

1 Introduction

Directed acyclic graph (DAG) models graphically represent conditional independencies among a set of random variables. For over fifteen years, decision analysts and computer scientists have used these models to encode the beliefs of experts (e.g., Howard & Matheson, 1981; Pearl, 1982; Heckerman & Wellman, 1995). More recently, statisticians and computer scientists have used these models for statistical inference or *learning* from data (e.g., Pearl & Verma, 1991; Cooper & Herskovits, 1992; Spirtes, Glymour, & Scheines, 1993; Spiegelhalter, Dawid, Lauritzen, & Cowell, 1993; Buntine, 1994; and Heckerman, Geiger, & Chickering, 1995). In particular, these researchers have applied model selection and model averaging techniques to the class of DAG models for the purposes of prediction and identifying cause and effect from observational data. The basic idea behind these endeavors has been that many domains exhibit conditional independence (e.g., due to causal relationships) and that DAG models are useful for capturing these relationships.

In this paper, we consider mixtures of DAG models (MDAG models) and Bayesian methods for learning models in this class. MDAG models generalize DAG models, and should more accurately model domains containing multiple distinct populations. In general, our hope is that the use of MDAG models will lead to better predictions and more accurate insights into causal relationships. In this paper, we concentrate on prediction.

The learning methods we consider are geared toward applications characterized by an extremely large number of possible models, large sample sizes, and the need for fast prediction. One example, which we shall consider in some detail, is the real-time compression of handwritten digits. For this class of application, the large number of possible models prohibits model averaging. Furthermore, the need for fast predictions prohibits the use of MCMC methods that average over a large number of possible models. Consequently, we concentrate on the task of selecting a single model among those possible. We use model posterior probability as our selection criterion.

For the class of applications we consider, even the task of model selection presents a computational challenge. In particular, a simple search-and-score approach is intractable, because methods for computing the posterior probability of an MDAG model, including Monte-Carlo and large-sample approximations, are extremely slow. In the paper, we introduce a heuristic method for MDAG model selection that addresses this difficulty. The method is not guaranteed to find the MDAG model with the highest probability, but experiments that we present suggest that it often identifies a good one. Our approach handles missing data and component DAG models that contain hidden or latent variables. Our approach can be used to learn DAG models (single-component MDAG models) from incomplete data as well.

Our method is based on two observations. One, an MDAG model can be viewed as a model containing (at least one) hidden variable. In particular, consider a domain containing random variables $\mathbf{X} = (X_1, \ldots, X_n)$ and a discrete random variable C in which the conditional distribution $p(\mathbf{x}|c)$ for each of the possible values c of C is encoded by a possibly different DAG model for $\mathbf{X}$. We call such a model a *multi-DAG model* for C and $\mathbf{X}$. If we marginalize over C, then $p(\mathbf{x})$ is given by the mixture of DAG models

$$p(\mathbf{x}) = \sum_c p(c)\ p(\mathbf{x}|c)$$

That is, a mixture of DAG models for $\mathbf{X}$ is a multi-DAG model for C and $\mathbf{X}$ where C is hidden (latent).

Two, there are efficient algorithms for selecting among multi-DAG models when data is complete—that is, when each sample contains observations for every random variable in the model, including C. These algorithms, which use heuristic search (e.g., greedy search), are straightforward generalizations of successful algorithms for selecting among DAG models given complete data (e.g., Cooper & Herskovits, 1992; Heckerman, Geiger, & Chickering, 1995). The algorithms for MDAG and DAG model selection are particularly efficient[1] because, given complete data, the posterior probability of a DAG model has a closed-form expression that factors into additive components for each node and its parents.

These observations suggest that, to learn an MDAG model for $\mathbf{X}$, one can augment the data (observations of subsets of $\mathbf{X}$) to include observations for all variables $\mathbf{X}$ and C, and apply an efficient algorithm for MDAG selection to the completed data. This strategy is the essence of our approach. To augment or complete the data, we compute expected sufficient statistics as is done in the Expectation–Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). Furthermore, we gradually improve data augmentation by interleaving search for parameters via the EM algorithm with structure search.

Our paper is organized as follows. In Section 2, we describe multi-DAG and MDAG models. In Section 3, we describe Bayesian methods for learning multi-DAG models, concentrating on the case where data is complete. In Section 4, we consider a simple approach for learning MDAGs that is computationally infeasible; and in Section 5, we modify the approach to produce a tractable class of algorithms. In Section 6, we evaluate the predictive accuracy of models produced by our approach using real examples. In Section 7, we describe a preliminary evaluation of our method for structure learning. Finally, in Sections 8 and 9, we describe related and future work, respectively.

[1] Throughout this paper, we use "efficiency" to refer to computational efficiency as opposed to statistical efficiency.

2 Multi-DAG models and mixtures of DAG models

In this section, we describe DAG, multi-DAG, and MDAG models. First, however, let us introduce some notation. We denote a random variable by an upper-case letter (e.g., X, Y, X_i, Θ), and the value of a corresponding random variable by that same letter in lower case (e.g., x, y, x_i, θ). When X is discrete, we use $|X|$ to denote the number of values of X, and sometimes refer to a value of X as a *state*. We denote a set of random variables by a bold-face capitalized letter or letters (e.g., $\mathbf{X}, \mathbf{Y}, \mathbf{Pa}_i$). We use a corresponding bold-face lower-case letter or letters (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{pa}_i$) to denote an assignment of value to each random variable in a given set. When $\mathbf{X} = \mathbf{x}$ we say that $\mathbf{X}$ is in *configuration* $\mathbf{x}$. We use $p(\mathbf{X} = \mathbf{x}|\mathbf{Y} = \mathbf{y})$ (or $p(\mathbf{x}|\mathbf{y})$ as a shorthand) to denote the probability or probability density that $\mathbf{X} = \mathbf{x}$ given $\mathbf{Y} = \mathbf{y}$. We also use $p(\mathbf{x}|\mathbf{y})$ to denote the probability distribution (both mass functions and density functions) for $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$. Whether $p(\mathbf{x}|\mathbf{y})$ refers to a probability, a probability density, or a probability distribution should be clear from context.

Suppose our problem domain consists of random variables $\mathbf{X} = (X_1, \ldots, X_n)$. A *DAG model for* $\mathbf{X}$ is a graphical factorization of the joint probability distribution of $\mathbf{X}$. The model consists of two components: a structure and a set of local distribution families. The structure $\mathbf{b}$ for $\mathbf{X}$ is a directed acyclic graph that represents conditional independence assertions through a factorization of the joint distribution for $\mathbf{X}$:

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}(\mathbf{b})_i) \tag{1}$$

where $\mathbf{pa}(\mathbf{b})_i$ is the configuration of the parents of X_i in structure $\mathbf{b}$ consistent with $\mathbf{x}$. The local distribution families associated with the DAG model are those in Equation 1. In this discussion, we assume that the local distribution families are parametric. Using $\boldsymbol{\theta}_b$ to denote the collective parameters for all local distributions, we rewrite Equation 1 as

$$p(\mathbf{x}|\boldsymbol{\theta}_b) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}(\mathbf{b})_i, \boldsymbol{\theta}_b) \tag{2}$$

With one exception to be discussed in Section 6, the parametric family corresponding to the variable X will be determined by (1) whether X is discrete or continuous and (2) the model structure. Consequently, we suppress the parametric family in our notation, and refer to the DAG model simply by its structure $\mathbf{b}$.

Let $\mathbf{b}^h$ denote the assertion or hypothesis that the "true" joint distribution can be represented by the DAG model $\mathbf{b}$ and has precisely the conditional independence assertions implied by $\mathbf{b}$. We find it useful to include the structure hypothesis explicitly in the factorization of the joint distribution when we compare model structures. In particular, we write

$$p(\mathbf{x}|\boldsymbol{\theta}_b, \mathbf{b}^h) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_b, \mathbf{b}^h) \tag{3}$$

This notation often makes it unnecessary to use the argument $\mathbf{b}$ in the term $\mathbf{pa}(\mathbf{b})_i$, and we use the simpler expression where possible.

The structure of a DAG model encodes a limited form of conditional independence that we call *context-non-specific conditional independence*. In particular, if the structure implies that two sets of random variables $\mathbf{Y}$ and $\mathbf{Z}$ are independent given some configuration of random variables $\mathbf{W}$, then $\mathbf{Y}$ and $\mathbf{Z}$ are also independent given every other configuration of $\mathbf{W}$. In a more general form of conditional independence, two sets of random variables may be independent given one configuration of $\mathbf{W}$, and dependent given another configuration of $\mathbf{W}$.

A multi-DAG model, called a Bayesian multinet by Geiger & Heckerman (1996), is a generalization of the DAG model that can encode context-specific conditional independence. In particular, *a multi-DAG model for* $\mathbf{X}$ *and distinguished random variable* C is a set of *component DAG models* for $\mathbf{X}$, each of which encodes the joint distribution for $\mathbf{X}$ given a state of C, and a distribution for C. Thus, the multi-DAG model for $\mathbf{X}$ and C encodes a joint distribution for $\mathbf{X}$ and C, and can encode context-specific conditional independence among these random variables, because the structure of each component DAG model may be different.

Let $\mathbf{s}$ and $\boldsymbol{\theta}_s$ denote the structure and parameters of a multi-DAG model for $\mathbf{X}$ and C. In addition, let $\mathbf{b}_c$ and $\boldsymbol{\theta}_c$ denote the structure and parameters of the cth DAG-model component of the multi-DAG model. Also, let $\mathbf{s}^h$ denote the hypothesis that the "true" joint distribution for $\mathbf{X}$ and C can be represented by the multi-DAG model $\mathbf{s}$ and has precisely the conditional independence assertions implied by $\mathbf{s}$. Then, the joint distribution for $\mathbf{X}$ and C encoded by this multi-DAG model is given by

$$p(c, \mathbf{x}|\boldsymbol{\theta}_s, \mathbf{s}^h) \;\;=\;\; p(c|\boldsymbol{\theta}_s, \mathbf{s}^h)\; p(\mathbf{x}|c, \boldsymbol{\theta}_s, \mathbf{s}^h) = \pi_c \; p(\mathbf{x}|\boldsymbol{\theta}_c, \mathbf{b}_c^h) \tag{4}$$

where $\boldsymbol{\theta}_s = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{|C|}, \pi_1, \ldots, \pi_{|C|})$ are the parameters of the multi-DAG model, $\pi_c = p(c|\boldsymbol{\theta}_s, \mathbf{s}^h)$, and $\mathbf{b}_c^h$ denotes the hypothesis that the "true" distribution for $\mathbf{X}$ given c can be encoded precisely using $\mathbf{b}_c$.

In what follows, we assume that the distinguished random variable has a multinomial distribution. In addition, with one exception to be discussed in Section 6, we limit the structure of the component DAG models and the parametric families for the local distributions as follows. When $X_i \in \mathbf{X}$ is a discrete random variable, we require that every random variable in $\mathbf{Pa}_i$ (for every component model) also be discrete, and that the local distribution families for X be a set of multinomial distributions, one for each configuration of $\mathbf{Pa}_i$. When $X_i \in \mathbf{X}$ is a continuous random variable, we require that the local distribution family for X_i be a set of linear-regressions over X_i's continuous parents with Gaussian error, one regression for each configuration of X_i's discrete parents. Lauritzen (1992) refers to this set of restrictions as a *conditional Gaussian distribution for a DAG model.*

In this paper, we concentrate on the special case where the distinguished random variable C is hidden. In this situation, we are interested in the joint distribution for $\mathbf{X}$, given by

$$p(\mathbf{x}|\boldsymbol{\theta}_s, \mathbf{s}^h) = \sum_{c=1}^{|C|} \pi_c \; p(\mathbf{x}|\boldsymbol{\theta}_c, \mathbf{b}_c^h) \tag{5}$$

This joint distribution is a mixture of distributions determined by the component DAG models, and has mixture weights $\pi_1, \ldots, \pi_{|C|}$. Thus, when C is hidden, we say that the multi-DAG model for $\mathbf{X}$ and C is a *mixture of DAG models (or MDAG model) for* $\mathbf{X}$.

An important subclass of DAG models is the *Gaussian DAG model* (e.g., Shachter & Kenley, 1989). In this subclass, the local distribution family for every random variable given its parents is a linear regression with Gaussian noise. It is well known that a Gaussian DAG model for $X_1, \ldots, X_n$ uniquely determines a multivariate-Gaussian distribution for those random variables. In such a model, the structure of the DAG model (in part) determines the "shape" of the multivariate-Gaussian distribution. Thus, the MDAG model class includes mixtures of multivariate-Gaussian distributions in which each component may have a different shape.

3 Learning multi-DAG models

In this and the following two sections, we consider a Bayesian approach for learning multi-DAG models and MDAG models. Let us assume that our data is exchangeable so that we can reason as if the data is a random sample from a true joint distribution. In addition, let us assume that the true joint distribution for $\mathbf{X}$ is encoded by some multi-DAG model, and that we are uncertain about both its structure and parameters. We define a discrete random variable $\mathbf{S}^h$ whose states $\mathbf{s}^h$ correspond to the possible true model hypotheses, and encode our uncertainty about structure using the probability distribution $p(\mathbf{s}^h)$. In addition, for each model $\mathbf{s}$, we define a continuous vector-valued random variable Θ_s, whose configurations $\boldsymbol{\theta}_s$ correspond to the possible true parameters. We encode our uncertainty about Θ_s using the probability density function $p(\boldsymbol{\theta}_s|\mathbf{s}^h)$.

Given a random sample $\mathbf{d} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ from the true distribution for $\mathbf{X}$, we compute the posterior distributions for each $\mathbf{s}^h$ and $\boldsymbol{\theta}_s$ using Bayes' rule:

$$p(\mathbf{s}^h|\mathbf{d}) = \frac{p(\mathbf{s}^h)\ p(\mathbf{d}|\mathbf{s}^h)}{\sum_{\mathbf{s}^h} p(\mathbf{s}^h)\ p(\mathbf{d}|\mathbf{s}^h)} \qquad\qquad p(\boldsymbol{\theta}_s|\mathbf{d}, \mathbf{s}^h) = \frac{p(\boldsymbol{\theta}_s|\mathbf{s}^h)\ p(\mathbf{d}|\boldsymbol{\theta}_s, \mathbf{s}^h)}{p(\mathbf{d}|\mathbf{s}^h)}$$

where

$$p(\mathbf{d}|\mathbf{s}^h) = \int p(\mathbf{d}|\boldsymbol{\theta}_s, \mathbf{s}^h)\ p(\boldsymbol{\theta}_s|\mathbf{s}^h)\ d\boldsymbol{\theta}_s \tag{6}$$

is the *marginal likelihood* of the model.

We can use the model posterior probability for various forms of model comparison, including model averaging (e.g., Bernardo & Smith, 1994). In this work, we limit ourselves to the selection of a model with a high posterior probability for reasons discussed in Section 1. In what follows, we concentrate on model selection using the posterior model probability. To simplify the discussion, we assume that all possible model structures are equally likely, a priori, in which case our selection criterion is the marginal likelihood.

3.1 The marginal likelihood criterion

Consider a DAG model $\mathbf{b}$ that encodes a conditional Gaussian distribution for $\mathbf{X}$. Let $\Theta_i, i = 1, \ldots, n$ denote the random variables corresponding to the parameters of the local distribution family for X_i. Buntine (1994) and Heckerman and Geiger (1995) have shown that, if (1) the parameter sets $\Theta_1, \ldots, \Theta_n$ are mutually independent given $\mathbf{b}^h$, (2) each of these parameter sets have a conjugate prior, and (3) the data $\mathbf{d}$ is complete for $\mathbf{X}$, then the log marginal likelihood has a closed form that can be computed efficiently.

This observation extends to multi-DAG models. Let Θ_{ic} denote the set of random variables corresponding to the local distribution family of X_i in component c. Also, let Π denote the set of random variables $(\Pi_1, \ldots, \Pi_{|C|-1})$ corresponding to the mixture weights. If (1) Π, Θ_{11}, $\ldots$, Θ_{n1}, $\ldots$, $\Theta_{1|C|}$, $\ldots$, $\Theta_{n|C|}$ are mutually independent given $\mathbf{s}^h$, (2) the parameter priors $p(\Theta_{ic}|\mathbf{s}^h)$ are conjugate for all i and c, and (3) the data $\mathbf{d}$ is complete for C and $\mathbf{X}$, then the marginal likelihood $p(\mathbf{d}|\mathbf{s}^h)$ has a closed form. In particular,

$$\log p(\mathbf{d}|\mathbf{s}^h) = \log p(\mathbf{d}^C) + \sum_{c=1}^{|C|} \log p(\mathbf{d}^{\mathbf{X},C=c}|\mathbf{b}_c^h) \tag{7}$$

where $\mathbf{d}^C$ is the data restricted to the random variable C, and $\mathbf{d}^{\mathbf{X},C=c}$ is the data restricted to the random variables $\mathbf{X}$ and those cases in which $C = c$. The term $p(\mathbf{d}^C)$ is the marginal likelihood of a trivial DAG model having only a single discrete node C. The terms in the sum are log marginal likelihoods for the component DAG models of the multi-DAG. Hence, $p(\mathbf{d}|\mathbf{s}^h)$ has a closed form.

3.2 Structure search

An important issue regarding model selection is the search for models (structures) with high posterior probabilities. Consider the problem of finding the DAG model with the highest marginal likelihood from the set of all models in which each node has no more than k parents. Chickering (1996) has shown the problem for $k > 1$ is NP-hard. It follows immediately that the problem of finding the multi-DAG model with the highest marginal likelihood from the set of all multi-DAGs in which each node in each component has no more than k parents is

NP-hard. Consequently, researchers use heuristic search algorithms including greedy search, greedy search with restarts, best-first search, and Monte-Carlo methods.

One consolation is that various model-selection criteria, including log marginal likelihood (under the assumptions just described), are factorable. We say that a criterion crit$(\mathbf{s}, \mathbf{d})$ for a multi-DAG structure $\mathbf{s}$ is *factorable* if it can be written as follows:

$$\text{crit}(\mathbf{s}, \mathbf{d}) = f(\mathbf{d}^C) + \sum_{c=1}^{|C|} \sum_{i=1}^{n} g_c(\mathbf{d}^{X_i, \mathbf{Pa}_i^c}) \tag{8}$$

where $\mathbf{d}^C$ is the data restricted to the set C, $\mathbf{Pa}_i^c$ are the parents of X_i in component c, $\mathbf{d}^{X_i, \mathbf{Pa}_i^c}$ is the data restricted to the random variables X_i and $\mathbf{Pa}_i^c$ and to those cases in which $C = c$, and f and g_c are functions. When a criterion is factorable, search is more efficient for two reasons. One, the component DAG models have non-interacting subcriteria so that we may search for a good DAG structure for each component separately. Two, as we search for a good structure in any one component, we need not reevaluate the criterion for the whole component. For example, in a greedy search for a good DAG structure, we iteratively transform the graph by choosing the transformation that improves the model criterion the most, until no such transformation is possible. Typical transformations include the removal, reversal, and addition of an arc (constrained so that the resulting graph is acyclic). Given a factorable criterion, we only need to reevaluate g_c for X_i if it's parents have changed.

4 Selecting among MDAGs: A simple approach

When learning multi-DAG models given complete data, the marginal likelihood has a closed form. In contrast, when learning MDAGs, the assumption that data is complete does not hold, because the distinguished random variable C is hidden. When data is incomplete, no tractable closed form for marginal likelihood is available. Nonetheless, we can approximate the marginal likelihood using either Monte-Carlo or large-sample methods (e.g., DiCiccio, Kass, Raftery, and Wasserman, 1995). Thus, a straightforward class of algorithm for choosing an MDAG model is to search among structures as before (e.g., perform greedy search), using some approximation for marginal likelihood. We shall refer to this class as *simple search-and-score algorithms.*

As we shall see, simple search-and-score algorithms for MDAG model selection are computationally infeasible in practice. Nonetheless, let us consider one approximation for the marginal likelihood that will help motivate a tractable class of algorithms that we consider in the next section. The approximation that we examine is a large-sample approximation first proposed by Cheeseman & Stutz (1995). Here, we develop the approximation using a heuristic argument given in Chickering & Heckerman (1997).

Let $\mathbf{d} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$, where $\mathbf{y}_i$ is the configuration of the observed variables in case i. As in Dempster et al. (1977), let $\mathbf{x}_i$ denote the ith *complete case*—the configuration of $\mathbf{X}$ and C in the ith case. Consider the identity

$$p(\mathbf{d}|\mathbf{s}^h) = p(\mathbf{d}_c|\mathbf{s}^h) \frac{\int p(\mathbf{d}, \boldsymbol{\theta}_s|\mathbf{s}^h)\ d\boldsymbol{\theta}_s}{\int p(\mathbf{d}_c, \boldsymbol{\theta}_s|\mathbf{s}^h)\ d\boldsymbol{\theta}_s} \tag{9}$$

where $\mathbf{d}_c = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ is any completion of the data set $\mathbf{d}$. Because $\mathbf{d}_c$ is a complete data set, we can determine $p(\mathbf{d}_c|\mathbf{s}^h)$ in closed form. Now, suppose we use Laplace's method to approximate the numerator and denominator of the second term. Roughly speaking, the resulting approximation for $p(\mathbf{d})$ will be best if the quantities $p(\mathbf{d}, \boldsymbol{\theta}_s|\mathbf{s}^h)$ and $p(\mathbf{d}_c, \boldsymbol{\theta}_s|\mathbf{s}^h)$—regarded as functions of $\boldsymbol{\theta}_s$—are similar in shape, so that errors in the two Laplace approximations tend to cancel (Tierney & Kadane, 1986). The two functions cannot be similar in an absolute sense, because $\mathbf{d}_c$ contains more information than does $\mathbf{d}$, and hence $p(\mathbf{d}_c, \boldsymbol{\theta}_s|\mathbf{s}^h)$ will be more peaked than $p(\mathbf{d}, \boldsymbol{\theta}_s|\mathbf{s}^h)$. Nonetheless, we can make the two functions more similar by completing $\mathbf{d}_c$ so that they peak for the same configuration of $\boldsymbol{\theta}_s$. That is, we want $\tilde{\boldsymbol{\theta}}_s^c$, the MAP configuration of $\boldsymbol{\theta}_s$ given $\mathbf{d}_c$, to be equal to $\tilde{\boldsymbol{\theta}}_s$, the MAP configuration given $\mathbf{d}$. One way to obtain this equality is to choose $\mathbf{d}_c$ so that its sufficient statistics match the expected sufficient statistics given $\mathbf{d}$ and $\mathbf{s}^h$. That is, we choose a particular completion $\mathbf{d}_c^*$ for $\mathbf{d}_c$ such that

$$\mathbf{t}_s(\mathbf{d}_c^*) = E_{\mathbf{d}_c|\mathbf{d},\tilde{\boldsymbol{\theta}}_s,\mathbf{s}^h}(\mathbf{t}_s(\mathbf{d}_c)) = \sum_{\mathbf{d}_c \in \mathcal{D}_c(\mathbf{d})} p(\mathbf{d}_c|\mathbf{d},\tilde{\boldsymbol{\theta}}_s,\mathbf{s}^h)\ \mathbf{t}_s(\mathbf{d}_c) \tag{10}$$

where $\mathbf{t}_s(\mathbf{d}_c)$ are the sufficient statistics for complete data $\mathbf{d}_c$ and model $\mathbf{s}$ and $\mathcal{D}_c(\mathbf{d})$ is the set of all possible completions of $\mathbf{d}$.[2] Note that the choice for $\mathbf{d}_c^*$ is computationally desirable because—when using the EM algorithm to find $\tilde{\boldsymbol{\theta}}_s$—the expected sufficient statistics are computed in the last expectation step.

If we set $\tilde{\boldsymbol{\theta}}_s^c = \tilde{\boldsymbol{\theta}}_s$ (as implied by our completion), apply Laplace's method to the numerator and denominator of Equation 9, and keep only terms that increase with the sample size N, we obtain

$$p(\mathbf{d}|\mathbf{s}^h) \approx p(\mathbf{d}_c^*|\mathbf{s}^h)\ \frac{p(\mathbf{d}|\tilde{\boldsymbol{\theta}}_s,\mathbf{s}^h)\ N^{d/2}}{p(\mathbf{d}_c^*|\tilde{\boldsymbol{\theta}}_s,\mathbf{s}^h)\ N^{d_c^*/2}} \tag{11}$$

where d and d_c^* are the dimensions of the model with respect to $\mathbf{d}$ and $\mathbf{d}_c^*$, respectively.[3] Equation 11 (without the correction for dimension) was introduced by Cheeseman & Stutz (1995) for use as a model-selection criterion in AutoClass. We sometimes refer to Equation 11 as the *Cheeseman–Stutz criterion*.

The approximation is a heuristic one. Nonetheless, in an empirical study with multinomial mixtures, Chickering & Heckerman (1997) have found the approximation to be quite good. In all experiments, this approximation was at least as accurate and sometimes more accurate than the standard approximation obtained using Laplace's method (e.g., Tierney & Kadane, 1986).

An important idea behind the Cheeseman–Stutz approximation is that we treat data completed by the EM algorithm as if it were real data. This same idea underlies the M step of the EM algorithm. As we shall see in the next section, this idea also can be applied to structure search.

5 Selecting among MDAGs: A practical approach

Consider a simple search-and-score approach for selecting MDAG models that uses the Cheeseman–Stutz criterion. This approach is inefficient, because the Cheeseman–Stutz criterion does not factor. Consequently, every time a transformation is applied to a structure during search, the entire structure may need to be rescored. This approach is also inefficient because the EM algorithm must be run every time a model is scored.

In this section, we introduce a modification of this approach that is efficient. They basic idea is that we modify the Cheeseman–Stutz approximation so that it does factor and so that EM need not be run every time a new model is scored. To accomplish the latter modification, we interleave parameter search (iterations of the EM algorithm) with structure search. The new algorithm is heuristic in that it is not guaranteed to find the MDAG model with the highest probability. As we shall see, however, it can work well in practice.

A sketch of the approach is shown in Figure 1. First, we choose some initial model $\mathbf{s}_0$ and parameter values $\boldsymbol{\theta}_0$. Then, we perform several iterations of the EM algorithm to find improved parameter values $\boldsymbol{\theta}_0^{'}$ for the structure. Next, we perform model search using a modification of the Cheeseman–Stutz criterion. The modified criterion factors and can be computed using only $(\mathbf{s}_0, \boldsymbol{\theta}_0^{'})$ and the data so that EM need not be run during search. The result of structure search is a new structure $\mathbf{s}_1$. Finally, we perform an M step to obtain parameters $\boldsymbol{\theta}_1$ for $\mathbf{s}_1$. We then

[2] The summation in Equation 10 denotes a generalized sum.

[3] In general, d will be less than or equal to d_c^*. See Geiger, Heckerman, & Meek (1996) for a discussion.

```
Choose initial structure $s_0$ and initial parameters $\theta_0$
Loop for $i = 0, 1, \ldots$ until convergence
    Run EM, using structure $s_i$ and initialization $\theta_i$, producing $\theta_i^{'}$
    Perform structure search, using modified criterion, producing $s_{i+1}$
    Perform M-step for $s_{i+1}$ producing $\theta_{i+1}$
```

Figure 1: A sketch of our approach for MDAG model selection.

iterate the EM, structure search, and M-step operations until some convergence criterion is satisfied.

As mentioned, our modification to the Cheeseman–Stutz criterion (Equation 11) is motivated by computational concerns. The modified criterion is

$$\text{criterion}(\mathbf{s}, \mathbf{d}) = \log p(\mathbf{d}_c^* | \mathbf{s}^h) \tag{12}$$

where $\mathbf{d}_c^*$ is chosen so that

$$\mathbf{t}_s(\mathbf{d}_c^*) = E_{\mathbf{d}_c | \mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i^h}(\mathbf{t}_s(\mathbf{d}_c)) = \sum_{\mathbf{d}_c \in \mathcal{D}_c(\mathbf{d})} p(\mathbf{d}_c | \mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i^h) \ \mathbf{t}_s(\mathbf{d}_c) \tag{13}$$

There are two differences between this approximation and the original one. One, we drop the "correction terms" $(p(\mathbf{d}|\tilde{\boldsymbol{\theta}}_s, \mathbf{s}^h) \ N^{d/2})/(p(\mathbf{d}_c^*|\tilde{\boldsymbol{\theta}}_s, \mathbf{s}^h) \ N^{d_c^*/2})$ from the Cheeseman–Stutz approximation so that the criterion factors. Experiments in Chickering and Heckerman (1996) involving multinomial mixture models suggest that dropping the correction terms does not significantly affect model selection. Two, we determine $\mathbf{d}_c^*$ using $(\mathbf{s}_i, \boldsymbol{\theta}_i^{'})$ rather than $(\mathbf{s}, \tilde{\boldsymbol{\theta}}_s)$ (where $\mathbf{s}$ is the current model under consideration) to avoid running EM during search. Although we know of no theoretical justification for this change, the hope is that $\mathbf{d}_c^*$ is a good approximation to the data completed using each of the models considered during structure search.

One computational difficulty remains, which we can address in a theoretically sound manner. In particular, the model criterion as we have described it requires that we compute expectations for every model we consider during structure search. We can replace these multiple evaluations with the computation of sufficient statistics for a single model as follows. First, note that we can determine the right-hand-side of Equation 12 using only the sufficient statistics of $\mathbf{d}_c^*$—we do not need the data set $\mathbf{d}_c^*$ itself. In addition, note that the sufficient statistics of the *complete model* (i.e., a model that encodes no conditional independence constraints) are sufficient statistics for any incomplete model. Consequently, we can determine the modified Cheeseman–Stutz criterion for every model considered during structure search by computing one set of expected sufficient statistics: the *expected complete model sufficient statistics*, given by

$$\text{ECMSS}(\mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i) = E_{\mathbf{d}_c | \mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i^h}(\mathbf{t}_{\text{comp}}(\mathbf{d}_c)) \tag{14}$$

We describe the computation of this (vector) quantity in the Appendix.

In the remainder of this section, we discuss variations on this approach, the initialization of both the structure and parameters, and a method for determining the number of mixture components and the number of states of any hidden variables in the component models.

One source of variation in our approach is the algorithm used for structure search once $\text{ECMSS}(\mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i)$ is computed. The options are the same as those for the simple search-and-score algorithms, and include greedy search, greedy search with restarts, best-first search, and Monte-Carlo methods. In preliminary studies, we have found greedy search to be effective; and in our analysis of real data in Section 6, we use this technique.

Another source of variation is the schedule used to alternate between parameter and structure search. With respect to parameter search, we can run EM to convergence, for one step, for some fixed number of steps, or for a number of steps that depends on how many times we have performed the search phase. With respect to structure search, we can perform model-structure transformations for some fixed number of steps, for some number of steps that depends on how many times we have performed the search phase, or until we find a local maximum. Finally, we can iterate the steps consisting of EM, the computation of ECMSS($\mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i$), and structure search until either (1) the MDAG structure does not change across two consecutive search phases, or (2) the approximate marginal likelihood of the resulting MDAG structure does not increase. Under the second schedule, the algorithm is guaranteed to terminate, because the marginal likelihood cannot increase indefinitely. Under the first schedule, we do not know of a proof that the algorithm will terminate. In our experiments with greedy structure search, however, we have found that this schedule halts.

We find it convenient to describe these schedules using a regular grammar, where E, M, E$_c$, S denote an E step, M step, computation of ECMSS($\mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i$), and structure search, respectively. For example, we use ((EM)*E$_c$S*M)* to denote the case where, within each outer iteration, we (1) run EM to convergence, (2) compute the expected complete model sufficient statistics, (3) run structure search to convergence, and (4) perform an M step. Another schedule we examine is ((EM)^{10}E$_c$S*M)*. In this schedule, we run EM for only 10 steps before computing the expected complete model sufficient statistics.[4]

We have evaluated various combinations of these schedules. Our experiments indicate that, although it is not necessary to run EM to convergence between structure search, a single EM step between structure searches selects models that have poor predictive accuracy. We have found that the schedule ((EM)^{10}E$_c$S*M)* works well for a variety of problems.

Our approach requires that both an initial structure and an initial parameterization be chosen. First, let us consider structural initialization. We initialize the structure of each component model by placing an arc from every hidden variable to every observable variable, with the exception that nodes corresponding to continuous random variables do not point to nodes corresponding to discrete random variables. A simpler choice for an initial graph is one in which every component consists of an empty graph—that is, a graph containing no arcs. However, with such an initialization and for a restricted set of priors, we conjecture that our approach would be unable to discover connections between hidden and observable variables.

Next, let us consider parameter initialization. When the mixture components contain no hidden continuous variables, we initialize parameters for a component DAG structure $\mathbf{b}$ as follows. First, we remove all hidden nodes and adjacent arcs from $\mathbf{b}$, creating model $\mathbf{b}'$. Next, we determine $\tilde{\boldsymbol{\theta}}_{b'}$, the MAP configuration for $\boldsymbol{\theta}_{b'}$ given data $\mathbf{d}$. Because the data is complete with respect to $\mathbf{b}'$, we can compute this MAP in closed form. Then, we create a conjugate distribution for $\boldsymbol{\theta}_{b'}$ whose configuration of maximum value agrees with the MAP configuration just computed and whose equivalent sample sizes are specified by the user. Next, for each non-hidden node X_i in $\mathbf{b}$ and for each configuration of X_i's hidden discrete parents, we initialize the parameters of the local distribution family for X_i by drawing from the conjugate distribution just described. For each hidden discrete node X_i in $\mathbf{b}$ and for each configuration of X_i's (possible) parents, we initialize the multinomial parameters associated with the local distribution family of X_i to be some fixed distribution (e.g., uniform). When the mixture components contain hidden continuous variables, we use the simpler approach of initializing parameters at random (i.e., by drawing from a distribution such as the prior). Methods for initializing the parameters of the distinguished random variable C include (1) setting the parameters to be equal, (2) setting the parameters to their prior means, and (3) drawing the parameters from a Dirichlet distribution.

Finally, the algorithm as described can compare neither models that contain different random variables nor models in which the same random variable has a different number of states.

[4] When the structure search leaves the model structure unchanged, we force another iteration of the outer loop in which we run EM to convergence rather than for 10 steps. If the model structure changes in this forced iteration, we continue to iterate with 10 EM steps.

Nonetheless, we can perform an additional search over the number of states of each discrete hidden variable by applying the algorithm in Figure 1 to initial models with different numbers of states for the hidden variables. We can discard a discrete hidden variable from a model by setting its number of states to one. After the best MDAG for each initialization is identified, we select the overall best structure using some criterion. Because only a relatively small number of alternatives are considered, we can use a computationally expensive approximation for the marginal likelihood such as the Cheeseman-Stutz approximation or a Monte-Carlo method.

6 Examples

In this section, we evaluate the predictive performance of MDAG models on real data. In addition, we evaluate some of the assumptions underlying our method for learning these models. In the domains that we consider, all the observable random variables are continuous. Consequently, we focus our attention on mixtures of Gaussian DAG models. To accommodate the outliers contained in the data sets that we analyze, each of the mixture models that we consider has a noise component in addition to one or more Gaussian components. The noise component is modeled as a multivariate uniform distribution, and can be viewed as an empty DAG model in which the distribution function for each of the random variables is uniform.

We compare the predictive performance of (1) mixtures of DAG models (MDAG/n) (2) mixtures of multivariate-Gaussian distributions for which the covariance matrices are diagonal (MDIAG/n), and (3) mixtures of multivariate-Gaussian distributions for which the covariance matrices are full (MFULL/n). The MDIAG/n and MFULL/n model classes correspond to MDAG models with fixed empty structures and fixed complete structures, respectively, for all Gaussian components. The /n suffix indicates the existence of a uniform noise component.

For every data set and every model class, we perform an outer search to identify the number of components within each mixture model as described in Section 5. In particular, we first learn a two-component model (one Gaussian and one noise component), and then increase by one the number of Gaussian mixture components until the model score is clearly decreasing. We choose the best number of components using the Cheeseman–Stutz criterion. Then, we measure the predictive ability of the chosen model **s** using the logarithmic scoring rule of Good (1952):

$$\frac{1}{|\mathbf{d}_{\text{test}}|} \sum_{l \in \mathbf{d}_{\text{test}}} \log p(\mathbf{x}_l | \mathbf{d}, \mathbf{s}^h) \tag{15}$$

where $\mathbf{d}_{\text{test}}$ is a set of test cases and $|\mathbf{d}_{\text{test}}|$ is the number of test cases. For mixture models with more than one component, we approximate the terms in this sum using a Gibbs sampler.

When learning MDAG/n models, we use the $((\mathrm{EM})^{10}\mathrm{E}_c\mathrm{S}^*\mathrm{M})^*$ search schedule; and when learning MDIAG/n and MFULL/n models, we run the EM algorithm to convergence. In all experiments, we deem EM to have converged when the the ratio of the change in log likelihood from the proceeding step and the change in log likelihood from the initialization falls below 10^{-6}. We initialize structure and parameters for our search procedures as described in Section 5 with equivalent sample sizes equal to 200.

6.1 Handwritten Digits

Our first example addresses the digital encoding of handwritten digits (Hinton, Dayan, & Revow, 1997). In this domain, there are 64 random variables corresponding to the gray-scale values [0,255] of scaled and smoothed 8-pixel x 8-pixel images of handwritten digits obtained from the CEDAR U.S. postal service database (Hull, 1994). Applications of joint prediction include image compression and digit classification. The sample sizes for the digits ("0" through "9") range from 1293 to 1534. For each digit, we use 1100 samples for training, and the remaining samples for testing.

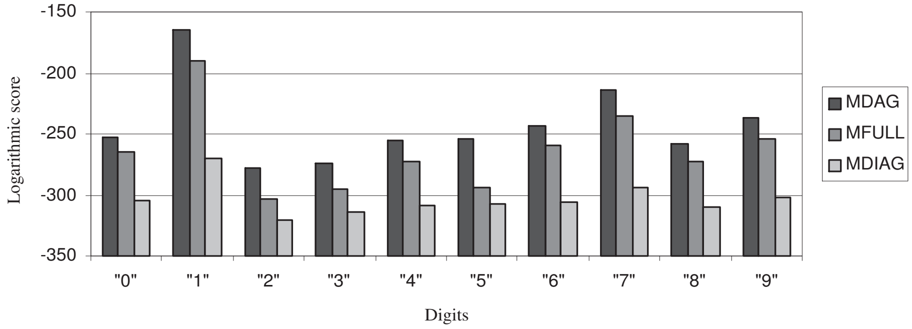

Figure 2: Logarithmic predictive scores on the test sets for the digit data.

	MDAG/n		MFULL/n		MDIAG/n	
Digit	*k*	*d*	*k*	*d*	*k*	*d*
“0”	5	1812	2	4290	8	1032
“1”	7	2910	2	4290	5	645
“2”	6	1816	1	2145	6	774
“3”	4	1344	1	2145	6	774
“4”	7	2115	2	4290	6	774
“5”	9	2702	1	2145	6	774
“6”	9	2712	2	4290	5	645
“7”	9	3168	2	4290	4	516
“8”	6	1868	2	4290	5	645
“9”	8	2955	2	4290	5	645

Table 1: Number of Gaussian components and parameters in the learned models for the digit data.

We use a relatively diffuse Normal-Wishart parameter prior for each of the Gaussian components of MFULL/n models. In the notation of DeGroot (1970), our prior has $\nu = 2$, all values in $\boldsymbol{\mu}$ set to 64 as a rough assessment of the average gray-scale value over pixels, $\alpha = \nu + 64$, and $\boldsymbol{\tau}$ set to the identity matrix. We choose α to be the sum of ν and the number of observed variables to compute the MAP parameter values in closed form. The parameter priors for the Gaussian components of the MDIAG/n and MDAG/n models are derived from the Normal-Wishart prior used in the MFULL/n models and the assumption of parameter modularity as described in Heckerman and Geiger (1995). We use a uniform prior on the number of components in the mixture and, when learning MDAG/n models, a uniform prior on the structure of the component DAG models. Because we know that the values of each of the 64 variables are constrained to the range [0,255], we fix the parameters in the uniform distribution of the noise model accordingly. Finally, the hyperparameters $\{\alpha_0, \ldots, \alpha_k\}$ of the Dirichlet prior on the mixture weights (i.e., the distinguished variable) are $\alpha_0 = 0.01$ for the noise component, and $\alpha_1 = \ldots = \alpha_k = 0.99/k$ for the Gaussian components.

The predictive logarithmic score on the test set for each digit is shown in Figure 2. The number of Gaussian components k and the model dimension d for the best model in each class are displayed in Table 1. Figure 2 indicates that MDAG/n models, on average, improve the predictive accuracy by 8% over MFULL/n models and 20% over MDIAG/n models. Note that the gains in predictive accuracy over MFULL/n models are obtained while reducing the average number of parameters by one third.

Using a P6 200MHz computer, the time taken to learn the MDAG/n, MFULL/n, and MDIAG/n models for a single digit—including the time needed to find the optimal number of components—is, on average, 6.0, 1.5, and 1.9 hours, respectively. These times could be improved by using a more clever search for the optimal number of mixture components.

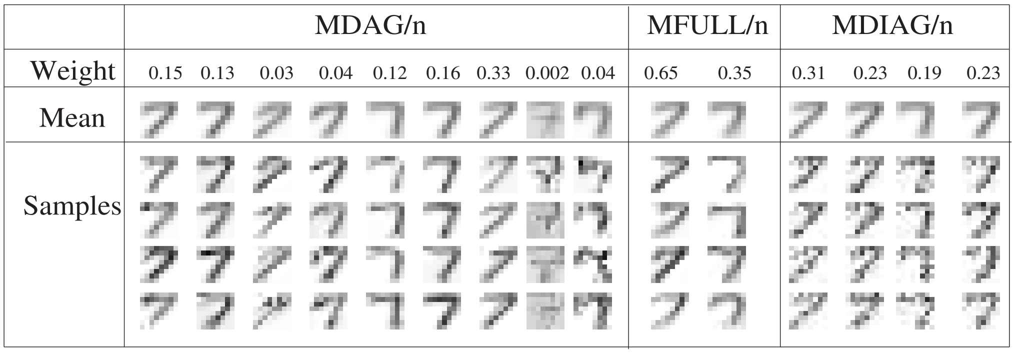

Figure 3: Means and samples from the components of the learned MDAG/n, MFULL/n, and MDIAG/n models for digit "7".

To better understand the differences in the distributions that these mixture models represent, we examine the individual Gaussian components for the learned MDAG/n, MFULL/n, and MDIAG/n models for the digit "7". The first row of Figure 3 shows the means for each of the components of each of the models. The mean values for the variables in each component are displayed in an 8 x 8 grid in which the shade of grey indicates the value of the mean. The displays indicate that each of the components of each type of model are capturing distinctive types of sevens. They do not, however, reveal any of the dependency structure in the component models. To help visualize these dependencies, we drew four samples from each component for each type of model. The grid for each sample is shaded to indicate the sampled values. Whereas the samples from the MDIAG/n components do look like sevens, they are mottled. This is not surprising, because each of the variables are conditionally independent given the component. The samples for the MFULL/n components are not mottled, but indicate that multiple types of sevens are being represented in one component. That is, several of the samples look blurred and appear to have multiple sevens superimposed. Generally, samples from each MDAG/n component look like sevens of the same distinct style, all of which closely resemble the mean.

Let us turn our attention to the evaluation of one of the key assumptions underlying our learning method. As we have discussed, the criterion used to guide structure search (Equation 12) is only a heuristic approximation to the true model posterior. To investigate the quality of this approximation, we can evaluate the model posterior using the Cheeseman-Stutz approximation (what we believe to be a more accurate approximation) for intermediate models visited during structure search. If the heuristic criterion is good, then the Cheeseman–Stutz criterion should increase as structure search progresses. We perform this evaluation when learning a three-component MDAG model for the digit "7" using the $((\mathrm{EM})^{10}\mathrm{E}_c\mathrm{S}^*\mathrm{M})^*$ schedule. For 149 out of the 964 model transitions, the Cheeseman–Stutz approximation decreased. Overall, however, as shown in Figure 4, the Cheeseman–Stutz score progresses upward to apparent convergence. We obtain similar results for other data sets. These results suggest that the heuristic criterion (Equation 12) is a useful guide for structure search.

Using statistics from this same experiment, we are able to estimate the time it would take to learn the MDAG model using the simple search-and-score approach described in Section 4. We find that the time to learn the three-component MDAG model for the digit "7", using the Cheeseman–Stutz approximation for model comparison, is approximately 6 years on a P6 200MHz computer, thus substantiating our previous claim about the intractability of simple search-and-score approaches.

Finally, a natural question is whether the Cheeseman–Stutz approximation for the marginal likelihood is accurate for model selection. The answer is important, because the MDAG models we select and evaluate are chosen using this approximation. Some evidence for the reasonableness of the approximation is provided by the fact that, as we vary the number of components

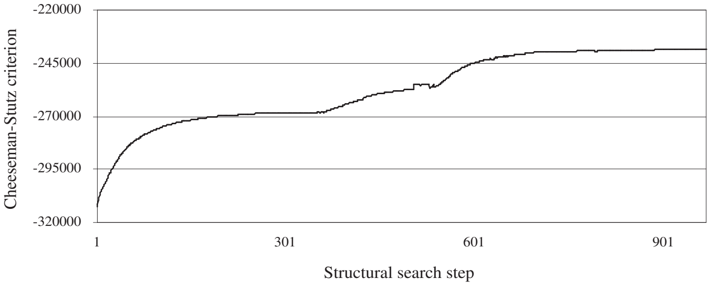

Figure 4: The Cheeseman–Stutz criterion for each intermediate model obtained during structure search when learning a three-component model for the digit “7”. The abrupt increases around steps 1, 350, and 540 occur when structure search transitions to a new component.

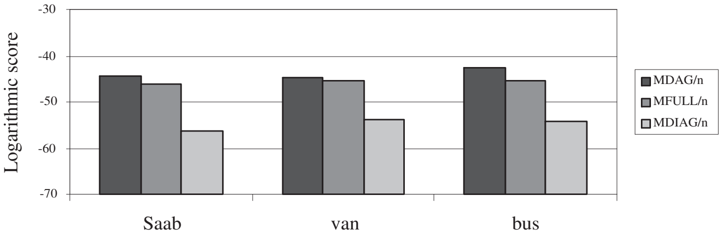

Figure 5: Logarithmic predictive scores on the test sets for the vehicle data.

of the MDAG models, the Cheeseman-Stutz and predictive scores roughly rise and fall in synchrony, usually peaking at the same number of components.

6.2 Vehicle Images

Our second example addresses the digital encoding of vehicle images. In this domain, there are 18 continuous features extracted from the silhouettes of four types of vehicles: a double decker bus (218 cases), a Saab 9000 (212 cases), an Opel Manta 400 (217 cases), and a Cheverolet van (199 cases). For each of the vehicles, we use 60% of the samples for training and 40% for testing. The data was obtained from the Machine Learning Repository (Siebert, 1987). As in the previous example, applications of joint prediction include image compression and classification.

The parameter priors for the Gaussian components, structure prior, and mixture priors are specified as described for the digits data set. For the variables in this data set, there are no natural constraints on the range of their values, so we use mutually independent bilateral bivariate Pareto priors for the independent uniform distributions within the noise component. In the notation of DeGroot (1970), the hyperparameters r_1 and r_2 for each of the 18 random variables are set using the minimum and maximum observed values for the Opel Manta. The hyperparameter α is set to 2 for each of these random variables.

The logarithmic predictive scores on the remaining types of vehicles are shown in Figure 5. For each of the vehicles, the learned MDAG/n models perform better in predictive scores than the other types of models. The improvements, however, are not as substantial as in the case of the digits data set.

7 Structure learning: A preliminary study

As we have mentioned in the introduction, many computer scientists and statisticians are using statistical inference techniques to learn the structure of DAG models from observational (i.e., non-experimental data). Pearl & Verma (1991) and Spirtes et al. (1993) have argued

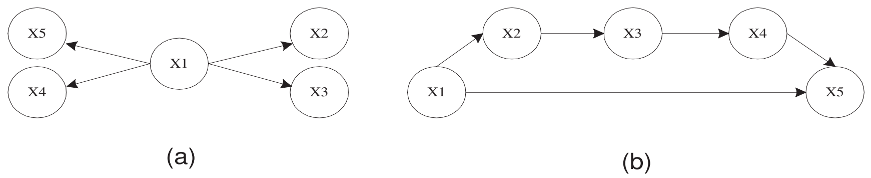

Figure 6: (a) The graphical structure for first and third components in the gold-standard MDAG. (b) The graphical structure for second component.

Sample		Weight of three	Arc differences		
size	k	largest comp.	COMP1	COMP2	COMP3
93	2	1.00	-	4	0
186	2	1.00	-	2	0
375	3	1.00	1	1	0
750	5	0.98	1	1	0
1500	3	1.00	0	3	0
3000	5	0.99	1	1	0

Table 2: Performance on the task of structure learning as a function of sample size.

that, under a set of simple (and sometimes reasonable) assumptions, the structures so learned can be used to infer cause-and-effect relationships. An interesting possibility is that these results can be generalized so that we may use the structure of learned MDAG models to infer causal relationships in mixed populations (populations in which subgroups have different causal relationships). In this section, we present a preliminary investigation into how well our approach can learn MDAG structure.

We perform our analysis as follows. First, we construct a "gold-standard" MDAG model, and use the model to generate data sets of varying size. Then, for each data set, we use our approach to learn an MDAG model (without a noise component). Finally, we compare the structure of the learned model to that of the gold-standard model, and measure the minimum number of arc manipulations (additions, deletions, and reversals) needed to transform each learned component structure to the corresponding gold-standard structure.

The gold-standard model is an MDAG model for five continuous random variables. The model has three mixture components. The structure of the first and third components (COMP1 and COMP3) are identical and this structure is shown in Figure 6a. The structure of the second component (COMP2) is shown in Figure 6b. The DAGs are parameterized so that there is some spatial overlap. In particular, all unconditional means in COMP1 and COMP2 are zero; all means in COMP3 are equal to five; and all linear coefficients and conditional variances are one (see Shachter & Kenley, 1989).

We construct a data set of size $N = 3000$ by sampling 1000 cases from each component of the gold-standard model. We then iteratively subsample this data, creating data sets of size $N = 93$, 186, 375, 750, 1500, and 3000.

Table 2 shows the results of learning models from the six data sets using the $((\mathrm{EM})^{10}\mathrm{E}_c\mathrm{S}^*\mathrm{M})^*$ schedule. The columns of the table contain the number of components k in the learned MDAG, the sum of the mixture weights in the three largest components and the minimum number of arc manipulations (additions, deletions, and reversals) needed to transform each learned component structure to the corresponding gold-standard structure for the three components with the largest mixture weights. Arc manipulations that lead to a model with different structures but the same family of distributions are not included in the count. All learned MDAG structures are close to that of the gold-standard model. In addition, although not apparent from the table, the structure of every learned component has only additional arcs in comparison with the gold-standard model for sample sizes larger than 375. Finally, it is interesting to note that, essentially, the structure is recovered for a sample size as low as 375.

8 Related work

DAG models (single-component MDAG models) with hidden variables generalize many well-known statistical models including linear factor analysis, latent factor models (e.g., Clogg, 1995), and probabilistic principle component analysis (Tipping & Bishop, 1997). MDAG models generalize a variety of mixtures models including naive-Bayes models used for clustering (e.g., Clogg, 1995; Cheeseman and Stutz, 1995), mixtures of factor analytic models (Hinton, Dayan, & Revow, 1997), and mixtures of probabilistic principle component analytic models (Tipping & Bishop, 1997).

Another interesting class of mixture models that has been considered can be obtained by enforcing equality constraints across the mixture components. Celeux & Govaert (1995) (extending the work of Banfield & Raftery, 1993) developed a class of models for continuous variables in which the covariance matrix of each component is reparameterized into a volume, a shape, and an orientation. In terms of mixtures of DAG models, these authors consider equality constraints on volume, shape, orientation, and size (magnitude of mixture weight), in various combinations for mixtures of complete models and empty models. Using MDAGs, it is possible to further extend the class of models developed by Banfield and Raftery. In particular, the size and volume of a multivariate-Gaussian distribution may be varied independently of the model structure. In contrast, the structure of a DAG model constrains the shape and orientation of the multivariate-Gaussian distributions. Thus, by considering MDAG models in which (1) the component structures are equal, and (2) we have equality constraints on various combinations of the size, volume, structure, shape, and orientation across components, we can capture the entire Banfield and Raftery hierarchy. Because the component model structures need not be restricted to complete or empty models, this MDAG hierarchy extends the original hierarchy.

There is also work related to our learning methods. The idea of interleaving parameter and structure search to learn graphical models has been discussed by Meilă, Jordan, & Morris (1997), Singh (1997), and Friedman (1997). Meilă et al. (1997) consider the problem of learning mixtures of DAG models for discrete random variables where each component is a spanning tree. Similar to our approach, they treat expected data as real data to produce a completed data set for structure search. Unlike our work, they replace heuristic model search with a polynomial algorithm for finding the "best" spanning-tree components given the completed data. Also, unlike our work, they use likelihood as a selection criterion, and thus do not take into account the complexity of the model.

Singh (1997) concentrates on learning a single DAG model for discrete random variables with incomplete data. He does not consider continuous variables or mixtures of DAG models. In contrast to our approach, Singh (1997) uses a Monte-Carlo method to produce completed data sets for structure search.

Friedman (1997, 1998) describes general algorithms for learning DAG models given incomplete data, and provides theoretical justification for some of his methods. Similar to our approach and the approach of Meilă et al. (1997), Friedman treats expected data as real data to produce completed data sets. In contrast to our approach, Friedman obtains the expected sufficient statistics for a new model using the current model. Most of these statistics are calculated by performing probabilistic inference in the current DAG model, although some of the statistics are obtained from a cache of previous inferences. In our approach, we only need to perform inference once on every case that has missing values to compute the expected complete model sufficient statistics. After these statistics are computed, model scores for arbitrary structures can be computed without additional inference.

9 Discussion and future work

We have described mixtures of DAG models, a class of models that is more general than DAG models, and have presented a novel heuristic method for choosing good models in this class. Although evaluations for more examples (especially ones containing discrete variables) are needed, our preliminary evaluations suggest that model selection within this expanded model class can lead to substantially improved predictions. This result is fortunate, as our evaluations also show that simple search-and-score algorithms, in which models are evaluated one at a time using Monte-Carlo or large-sample approximations for model posterior probability, are intractable for some real problems.

One important observation from our evaluations is that the (practical) selection criterion that we introduce—the marginal likelihood of the complete model sufficient statistics—is a good guide for model search. An interesting question is: Why? We hope that this work will stimulate theoretical work to answer this question and perhaps uncover better approximations for guiding model search. Friedman (1998) has some initial insight.

A possibly related challenge for theoretical study has to do with the apparent accuracy of the Cheeseman–Stutz approximation for the marginal likelihood. As we have discussed, in experiments with multinomial mixtures, Chickering & Heckerman (1997) have found the approximation to be at least as accurate and sometimes more accurate than the standard Laplace approximation. Our evaluations have also provided some evidence that the Cheeseman–Stutz approximation is an accurate criterion for model selection.

In our evaluations, we have not considered situations where the component DAG models themselves contain hidden variables. In order to learn models in this class, new methods for structure search are needed. In such situations, the number of possible models is significantly larger than the number of possible DAGs over a fixed set of variables. Without constraining the set of possible models with hidden variables—for instance, by restricting the number of hidden variables—the number of possible models is infinite. On a positive note, Spirtes et al. (1993) have shown that constraint-based methods under suitable assumptions can sometimes indicate the existence of a hidden common cause between two variables. Thus, it may be possible to use the constraint-based methods to suggest an initial set of plausible models containing hidden variables that can then be subjected to a Bayesian analysis.

In Section 7, we saw that we can recover the structure of an MDAG model to a fair degree of accuracy. This observation raises the intriguing possibility that we can infer causal relationships from a population consisting of subgroups governed by different causal relationships. One important issue that needs to be addressed first, however, has to do with structural identifiability. For example, two MDAG models may superficially have different structures, but may otherwise be statistically equivalent. Although the criteria for structural identifiability among single-component DAG models is well known, such criteria are not well understood for MDAG models.

Appendix: Expected complete model sufficient statistics

In this appendix, we examine complete model sufficient statistics more closely. We shall limit our discussion to multi-DAG models for which the component DAG models have conditional Gaussian distributions. The extension to the noise component is straightforward.

Consider a multi-DAG model for random variables C and $\mathbf{X}$. Let $\boldsymbol{\Gamma}$ denote the set of continuous variables in $\mathbf{X}$, $\boldsymbol{\gamma}$ denote a configuration of $\boldsymbol{\Gamma}$, and n_c denote the number of variables in $\boldsymbol{\Gamma}$. Let $\boldsymbol{\Delta}$ denote the set of all discrete variables (including the distinguished variable C), and m denote the number of possible configurations of $\boldsymbol{\Delta}$. As we have discussed, let $\mathbf{d} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ and $\mathbf{d}_c = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$.

Now, consider the complete model sufficient statistics for a complete case, which we denote

$\mathbf{t}_{\mathrm{comp}}(\mathbf{x})$. For the multi-DAG model, $\mathbf{t}_{\mathrm{comp}}(\mathbf{x})$ is a vector $\langle\langle N_1, R_1, S_1\rangle, \ldots, \langle N_m, R_m, S_m\rangle\rangle$ of m triples, where the N_j are scalars, the R_j are vectors of length n_c, the S_j are square matrices of size $n_c \times n_c$. In particular, if the discrete variables in $\mathbf{x}$ take on the j^{th} configuration, then $N_j = 1$, $R_j = \boldsymbol{\gamma}$, and $S_j = \boldsymbol{\gamma}' * \boldsymbol{\gamma}$, and $N_k = 0$, $R_k = 0$, and $S_k = 0$ for $k \neq j$.

For an incomplete data set, the expected complete model sufficient statistics is given by Equation 14. Assuming the data are exchangeable, we can compute these expected sufficient statistics as follows:

$$\mathrm{ECMSS}(\mathbf{d}, \boldsymbol{\theta}_i, \mathbf{s}_i) = \sum_{l=1}^{N} E_{\mathbf{d}_c|\mathbf{d},\boldsymbol{\theta}_i,\mathbf{s}_i^h}(\mathbf{t}_{\mathrm{comp}}(\mathbf{x}_l)) \tag{16}$$

The expectation of $\mathbf{t}_{\mathrm{comp}}(\mathbf{x})$ is computed by performing probabilistic inference in the multi-DAG model. Such inference is a simple extension of the work of Lauritzen (1992). The sum of expectations are simply scalar, vector, or matrix additions (as appropriate) in each triple in each of the coordinates of the vector.

Note that, in the computation as we have described it, we require a statistic triple for every possible configuration of discrete variables. In practice, however, we can use a sparse representation in which we store triples only for those complete observations that are consistent with the data.

References

Banfield, J. and Raftery, A. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821.

Bernardo, J. and Smith, A. (1994). *Bayesian Theory.* John Wiley and Sons, New York.

Buntine, W. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, Vol. 28, No. 5, pages 781–793.

Cheeseman, P. and Stutz, J. (1995). Bayesian classification (AutoClass): Theory and results. In Fayyad, U., Piatesky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI Press, Menlo Park, CA.

Chickering, D. (1996). Learning Bayesian networks is NP-complete. In Fisher, D. and Lenz, H., editors, *Learning from Data*, pages 121–130. Springer-Verlag.

Chickering, D. and Heckerman, D. (1996). Efficient approximations for the marginal likelihood of incomplete data given a Bayesian network. In *Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence,* Portland, OR, pages 158–168. Morgan Kaufmann.

Chickering, D. and Heckerman, D. (1997). Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29:181–212.

Clogg, C. (1995). Latent class models. In *Handbook of statistical modeling for the social and behavioral sciences*, pages 311–359. Plenum Press, New York.

Cooper, G. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

DeGroot, M. (1970). *Optimal Statistical Decisions.* McGraw-Hill, New York.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39:1–38.

DiCiccio, T., Kass, R., Raftery, A., and Wasserman, L. (July, 1995). Computing Bayes factors by combining simulation and asymptotic approximations. Technical Report 630, Department of Statistics, Carnegie Mellon University, PA.

Friedman, N. (1997). Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning.* Morgan Kaufmann, San Mateo, CA.

Friedman, N. (1998). The Bayesian structural EM algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence Learning.* Morgan Kaufmann, San Mateo, CA. To appear.

Geiger, D. and Heckerman, D. (1996). Beyond Bayesian networks: Similarity networks and Bayesian multinets. *Artificial Intelligence*, 82:45–74.

Geiger, D., Heckerman, D., and Meek, C. (1996). Asymptotic model selection for directed networks with hidden variables. In *Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence,* Portland, OR, pages 283–290. Morgan Kaufmann.

Good, I. (1952). Rational decisions. *J. R. Statist. Soc. B*, 14:107–114.

Heckerman, D. and Geiger, D. (1995). Likelihoods and priors for Bayesian networks. Technical Report MSR-TR-95-54, Microsoft Research, Redmond, WA.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243.

Heckerman, D. and Wellman, M. (1995). Bayesian networks. *Communications of the ACM*, 38:27–30.

Hinton, G., Dayan, P., and Revow, M. (1997). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8:65–74.

Howard, R. and Matheson, J. (1981). Influence diagrams. In Howard, R. and Matheson, J., editors, *Readings on the Principles and Applications of Decision Analysis*, volume II, pages 721–762. Strategic Decisions Group, Menlo Park, CA.

Hull, J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:550–554.

Lauritzen, S. (1992). Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87:1098–1108.

Meilă, M., Jordan, M., and Morris, Q. (1997). Estimating dependency structure as a hidden variable. Technical Report 1611, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings AAAI-82 Second National Conference on Artificial Intelligence,* Pittsburgh, PA, pages 133–136. AAAI Press, Menlo Park, CA.

Pearl, J. and Verma, T. (1991). A theory of inferred causation. In Allen, J., Fikes, R., and Sandewall, E., editors, *Proceedings of Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 441–452. Morgan Kaufmann, San Mateo, CA.

Shachter, R. and Kenley, C. (1989). Gaussian influence diagrams. *Management Science*, 35:527–550.

Siebert, J. (1987). Vehicle recognition using rule-based methods. Technical Report TIRM-87-018, Turing Institute.

Singh, M. (1997). Learning Bayesian networks from incomplete data. In *Proceedings AAAI-97 Fourteenth National Conference on Artificial Intelligence,* Providence, RI, pages 534–539. AAAI Press, Menlo Park, CA.

Spiegelhalter, D., Dawid, A., Lauritzen, S., and Cowell, R. (1993). Bayesian analysis in expert systems. *Statistical Science*, 8:219–282.

Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, Prediction, and Search.* Springer-Verlag, New York.

Tierney, L. and Kadane, J. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81:82–86.

Tipping, M. and Bishop, C. (1997). Mixtures of probabilistic principle component analysers. Technical Report NCRG-97-003, Neural Computing Research Group.