Machine Learning, 29, 181–212 (1997)

Efficient Approximations for the Marginal Likelihood of Bayesian Networks with Hidden Variables

DAVID MAXWELL CHICKERING dmax@microsoft.com
DAVID HECKERMAN heckerma@microsoft.com
Microsoft Research, Redmond, WA 98052-6399

Editor: Padhraic Smyth

Abstract. We discuss Bayesian methods for model averaging and model selection among Bayesian-network models with hidden variables. In particular, we examine large-sample approximations for the marginal likelihood of naive-Bayes models in which the root node is hidden. Such models are useful for clustering or unsupervised learning. We consider a Laplace approximation and the less accurate but more computationally efficient approximation known as the Bayesian Information Criterion (BIC), which is equivalent to Rissanen's (1987) Minimum Description Length (MDL). Also, we consider approximations that ignore some off-diagonal elements of the observed information matrix and an approximation proposed by Cheeseman and Stutz (1995). We evaluate the accuracy of these approximations using a Monte-Carlo gold standard. In experiments with artificial and real examples, we find that (1) none of the approximations are accurate when used for model averaging, (2) all of the approximations, with the exception of BIC/MDL, are accurate for model selection, (3) among the accurate approximations, the Cheeseman–Stutz and Diagonal approximations are the most computationally efficient, (4) all of the approximations, with the exception of BIC/MDL, can be sensitive to the prior distribution over model parameters, and (5) the Cheeseman–Stutz approximation can be more accurate than the other approximations, including the Laplace approximation, in situations where the parameters in the maximum a posteriori configuration are near a boundary.

Keywords: Bayesian model averaging, model selection, multinomial mixtures, clustering, unsupervised learning, Laplace approximation

1. Introduction

There is growing interest in methods for learning graphical models from data. In this paper, we consider Bayesian methods such as those reviewed in Heckerman (1995) and Buntine (1996). A key step in the Bayesian approach to learning graphical models is the computation of the *marginal likelihood* of a data set given a model. This quantity is the ordinary likelihood (a function of the data and the model parameters) averaged over the parameters with respect to their prior distribution. Given a *complete* data set—that is, a data set in which each sample contains observations for every variable in the model—the marginal likelihood can be computed in closed form under certain assumptions (e.g., Cooper & Herskovits, 1992; Heckerman & Geiger, 1995). In contrast, when observations are missing, including situations where some variables are *hidden* (i.e., never observed), the exact determination of the marginal likelihood is typically intractable (e.g., Cooper & Herskovits, 1992). Consequently, approximate techniques for computing the marginal likelihood are used.

One class of approximations that has received a great deal of attention in the statistics community is based on Monte-Carlo techniques. In theory, these approximations are known to converge to an accurate result. In practice, however, the amount of computer time needed for convergence can be enormous. An alternative class of approximations is based on the large-sample properties of probability distributions. This class also can be accurate under certain assumptions, and are typically more efficient[1] than Monte-Carlo techniques.

One large-sample approximation, known as a *Laplace approximation*, is widely used by Bayesian statisticians (Haughton, 1988; Kass, Tierney, & Kadane, 1988; Kass & Raftery, 1995). Although this approximation is efficient relative to Monte-Carlo methods, it has a computational complexity of $O(d^2N)$ or greater, where d is the dimension of the model and N is the sample size of the data. Consequently, the Laplace approximation can be a computational burden for large models.

In this paper, we examine other large-sample approximations that are more efficient than the Laplace approximation. These approximations include the Bayesian Information Criterion (BIC) (Schwarz, 1978), which is equivalent to Rissanen's (1987) Minimum-Description-Length (MDL) measure, diagonal and block diagonal approximations for the Hessian term in the Laplace approximation (Becker & LeCun, 1988; Buntine & Weigand, 1994), and an approximation suggested by Cheeseman and Stutz (1995).

Researchers have investigated the accuracy and efficiency of some of these approximations. For example, both theoretical and empirical studies have shown that the Laplace approximation is more accurate than is the BIC/MDL approximation (e.g., Draper, 1993; Raftery, 1994). Also, Becker and LeCun (1989) and MacKay (1992b) report successful and unsuccessful applications of the diagonal approximation, respectively, in the context of parameter learning for probabilistic neural-network models.

In this paper, we fill in some of the gaps that have been left by previous studies. We examine empirically the accuracy and efficiency of all approximations, comparing them to a Monte-Carlo gold standard. We do so using simple Bayesian networks for discrete variables that contain a single hidden variable. To our knowledge, this empirical study is the first that compares these approximations with a Monte-Carlo standard in the context of hidden-variable Bayesian networks, and the first that examines the accuracy of the Cheeseman–Stutz approximation.

Our study is motivated by a need for accurate and efficient methods for exploratory data analysis. One exploration task is *clustering*. For example, suppose we have repeated observations for the discrete variables $\mathbf{X} = (X_1, \ldots, X_n)$. One possible model for clustering these observations is shown in Figure 1. In this *naive-Bayes* model, a discrete hidden variable C renders the observations conditionally independent, and the joint distribution over $\mathbf{X}$ is given by a mixture of multinomial distributions

$$p(\mathbf{x}) = \sum_{j=1}^{r_c} p(C = c^j) \prod_{i=1}^{n} p(x_i | C = c^j), \tag{1}$$

where r_c is the number of states of the hidden variable C. Each state c^j of C corresponds to an underlying cluster or class in the data. Such models for clustering have been used by Cheeseman and Stutz (1995) in their system called AutoClass, and have been studied in

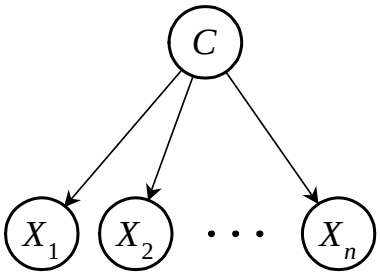

Figure 1. A Bayesian-network structure for clustering. The possible states of the hidden variable correspond to the underlying classes in the data.

depth by statisticians (e.g., Clogg, 1995). The approximations we examine can be used to determine the number of classes that is optimal according to the data (and prior information). Alternatively, we can use the approximations to provide weights for combining models with different numbers of classes.

Another important area of exploratory data analysis is causal discovery (Spirtes, Glymour, & Scheines, 1993), which can also be cast in terms of learning graphical models. Heckerman (1995) describes how approximations for the marginal likelihood can be used for this task.

In this paper, we seek to find one or more marginal-likelihood approximations for exploratory data analysis that are accurate and yet scale to large problems. We examine these approximations for the class of clustering models depicted in Figure 1. In Section 2, we review the basic Bayesian approach for model averaging and model selection, emphasizing the importance of the marginal likelihood. In Section 3, we describe Monte-Carlo and large-sample approximations for computing marginal likelihood when there is missing data. In Section 4, we evaluate the accuracy and efficiency of the various approximations, using a Monte-Carlo gold standard for comparison. We examine the approximations using both synthetic and real-world data.

2. Bayesian methods for learning: The basics

Commonly used Bayesian approaches for learning model structure include model averaging and model selection. These approaches date back to the work of Jeffreys (1939), and refinements can be found in (e.g.) Good (1965), Berger (1985), Gull and Skilling (1991), MacKay (1992a), Cooper and Herskovits (1992), Spiegelhalter, Dawid, Lauritzen, and Cowell (1993), Buntine (1994), Kass and Raftery (1995), and Heckerman, Geiger, and Chickering (1995). In this section, we review these methods and how they apply to learning with Bayesian networks given complete data.

First, we need some notation. We denote a variable by an upper-case letter (e.g., X, Y, X_i, Θ), and the state or value of a corresponding variable by that same letter in lower case (e.g., x, y, x_i, θ). We denote a set of 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 state or value to each variable in a given set. We say that variable set $\mathbf{X}$ 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.

Now, suppose our problem domain consists of variables $\mathbf{X} = (X_1, \ldots, X_n)$. In addition, suppose that we have some data $D = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, which is a random sample from some unknown probability distribution for $\mathbf{X}$. In this section, we assume that each case $\mathbf{x}$ in D consists of an observation of all the variables in $\mathbf{X}$. We assume that the unknown probability distribution can be encoded by some statistical model with structure $\mathbf{m}$ and parameters $\boldsymbol{\theta}_m$. We are uncertain about the structure and parameters of the model, and—using the Bayesian approach—we encode this uncertainty using probability. In particular, we define a discrete variable $\mathbf{M}$ whose states $\mathbf{m}$ correspond to the possible true models, and encode our uncertainty about $\mathbf{M}$ with the probability distribution $p(\mathbf{m})$. In addition, for each model structure $\mathbf{m}$, we define a continuous vector-valued variable Θ_m, whose configurations $\boldsymbol{\theta}_m$ correspond to the possible true parameters. We encode our uncertainty about Θ_m using the probability density function $p(\boldsymbol{\theta}_m|\mathbf{m})$.

Given random sample D, we compute the posterior distributions for each $\mathbf{m}$ and $\boldsymbol{\theta}_m$ using Bayes' rule:

$$p(\mathbf{m}|D) = \frac{p(\mathbf{m})\, p(D|\mathbf{m})}{\sum_{\mathbf{m}'} p(\mathbf{m}')\, p(D|\mathbf{m}')} \tag{2}$$

$$p(\boldsymbol{\theta}_m|D, \mathbf{m}) = \frac{p(\boldsymbol{\theta}_m|\mathbf{m})\, p(D|\boldsymbol{\theta}_m, \mathbf{m})}{p(D|\mathbf{m})} \tag{3}$$

where

$$p(D|\mathbf{m}) = \int p(D|\boldsymbol{\theta}_m, \mathbf{m})\, p(\boldsymbol{\theta}_m|\mathbf{m})\, d\boldsymbol{\theta}_m \tag{4}$$

is the *marginal likelihood*. Given some hypothesis of interest, h, we determine the probability that h is true given data D by averaging over all possible models and their parameters according to the rules of probability:

$$p(h|D) = \sum_{\mathbf{m}} p(\mathbf{m}|D)\, p(h|D, \mathbf{m}) \tag{5}$$

$$p(h|D, \mathbf{m}) = \int p(h|\boldsymbol{\theta}_m, \mathbf{m})\, p(\boldsymbol{\theta}_m|D, \mathbf{m}) d\boldsymbol{\theta}_m. \tag{6}$$

For example, h may be the event that the next observation is $\mathbf{x}_{N+1}$. In this situation, we obtain

$$p(\mathbf{x}_{N+1}|D) = \sum_m p(\mathbf{m}|D) \int p(\mathbf{x}_{N+1}|\boldsymbol{\theta}_m, \mathbf{m})\, p(\boldsymbol{\theta}_m|D, \mathbf{m}) d\boldsymbol{\theta}_m, \tag{7}$$

where $p(\mathbf{x}_{N+1}|\boldsymbol{\theta}_m, \mathbf{m})$ is the likelihood for the model. This approach is often referred to as *Bayesian model averaging*. Note that no single model structure is learned. Instead, all possible models are weighted by their posterior probability.

Model averaging is not always appropriate for an analysis. For example, only one or a few models may be desired for domain understanding or for fast prediction. In these situations, we select one or a few "good" model structures from among all possible models, and use them as if they were exhaustive. This procedure is known as *model selection* when one model is chosen and as *selective model averaging* when more than one model is chosen. Of course, model selection and selective model averaging are also useful when it is impractical to average over all possible model structures.

Whether a model is "good" will depend on the particular application. For example, a good model for understanding the causal relationships in a domain will not necessarily be a good model for a classification or regression task. Also, if a model is to be used for decision making, its quality will likely depend on the alternatives available and the preferences of the decision maker. These issues are discussed in more detail by (e.g.) Spiegelhalter et al. (1993) and Heckerman (1995). Nonetheless, the relative posterior probability of a model structure, $p(D, \mathbf{m}) = p(\mathbf{m})\ p(D|\mathbf{m})$, is often used as a general-purpose criterion for selective model averaging and model selection.[2] Consequently, the marginal likelihood is important for both model averaging and model selection.

Now let us assume that our statistical model is a Bayesian network. A *Bayesian network* for $\mathbf{X}$ consists of a directed-acylic-graph structure $\mathbf{m}$ and a set of *local distribution functions* $p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_m, \mathbf{m})$, where $\mathbf{Pa}_i$ is the set of variables that corresponds to the parents of X_i in the graph. The structure $\mathbf{m}$ encodes the independence assumptions

$$p(\mathbf{x}|\boldsymbol{\theta}_m, \mathbf{m}) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_m, \mathbf{m}). \tag{8}$$

Under certain assumptions, the computations needed for Bayesian model averaging, selective model averaging, or model selection can be done efficiently and in closed form. Many researchers who have addressed Bayesian-network learning have adopted at least some of these assumptions (e.g., Cooper & Herskovits, 1992; Spiegelhalter et al., 1993; Buntine, 1994; Heckerman et al., 1995). The assumptions include:

1. Every variable is discrete, having a finite number of states. We use x_i^k and $\mathbf{pa}_i^j$ to denote the kth possible state of X_i and the jth possible configuration of $\mathbf{Pa}_i$, respectively. Also, we use r_i and q_i to denote the number of possible states of X_i and the number of possible configurations of $\mathbf{Pa}_i$, respectively.

2. Each local distribution function $p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_m, \mathbf{m})$ consists of a set of multinomial distributions, one multinomial distribution for each i and j. That is,

$$p(x_i^k|\mathbf{pa}_i^j, \boldsymbol{\theta}_m, \mathbf{m}) = \theta_{ijk},$$

where the θ_{ijk} are parameters satisfying $\theta_{ijk} > 0$ for all i, j, and k, and $\sum_{k=1}^{r_i} \theta_{ijk} = 1$ for all i and j. For convenience, we introduce the set of nonredundant parameters $\boldsymbol{\theta}_{ij} = (\theta_{ij2}, \ldots, \theta_{ijr_i})$ for all i and j.

3. The parameter sets $\boldsymbol{\theta}_{ij}$ are mutually independent, so that

$$p(\boldsymbol{\theta}_m|\mathbf{m}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|\mathbf{m}).$$

4. Each parameter set $\boldsymbol{\theta}_{ij}$ has a Dirichlet distribution, giving

$$p(\boldsymbol{\theta}_{ij}|\mathbf{m}) = \mathrm{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1}, \ldots, \alpha_{ijr_i}) \propto \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1},$$

where hyperparameters $\alpha_{ijk} > 0$ for every i, j, and k.

5. The data set D is *complete*—that is, every variable is observed in every case of D.

Under these assumptions, the parameters remain independent given a random sample D that contains no missing observations, so that

$$p(\boldsymbol{\theta}_m|D, \mathbf{m}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|D, \mathbf{m}), \tag{9}$$

and the posterior distribution of each $\boldsymbol{\theta}_{ij}$ will have the Dirichlet distribution

$$p(\boldsymbol{\theta}_{ij}|D, \mathbf{m}) = \mathrm{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i}), \tag{10}$$

where N_{ijk} is the number of cases in D in which $X_i = x_i^k$ and $\mathbf{Pa}_i = \mathbf{pa}_i^j$. Note that the collection of counts N_{ijk} are sufficient statistics of the data for the model $\mathbf{m}$. In addition, we obtain the marginal likelihood

$$p(D|\mathbf{m}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{11}$$

where $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$ and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. (See Cooper and Herskovits (1992) for a derivation.)

These assumptions are restrictive; and there has been a great deal of recent work building on general results from Bayesian statistics to relax these assumptions. For example, Geiger and Heckerman (1994) discuss the case where variables are continuous and each local distribution function corresponds to ordinary linear regression; Buntine (1994) discusses the more general case where local distribution functions come from the exponential family; MacKay (1992a) and Gilks, Richardson, and Spiegelhalter (1996) relax the assumption of parameter independence using hierarchical models; and Buntine (1994), Azevedo-Filho and Shachter (1995), Heckerman (1995), and Gilks et al. (1996) have addressed the case where data are missing.

3. Methods for missing data

When observations for some variables are missing in the data, the parameters for a given model become dependent, and known closed-form methods cannot be used to determine marginal likelihood. Approximations for computing marginal likelihood include Monte-Carlo approaches such as Gibbs sampling and importance sampling (Neal, 1993; Chib, 1995; Raftery, 1996) and large-sample approximations (Kass et al., 1988; Kass & Raftery, 1995). As mentioned in the introduction, Monte-Carlo methods are accurate but typically inefficient, whereas large-sample methods are more efficient but known to be accurate only for large data sets. In this paper, we examine the accuracy and efficiency of large-sample methods using a Monte-Carlo approximation as a standard for comparison. In this section, we describe the approximations that we use.

We note that, when treating missing data, an important consideration is whether or not we can ignore the process by which observations are missed. For example, a missing datum in a drug study cannot be ignored if there is the possibility that—as a result of taking the drug—the patient became too ill to be observed. In contrast, if data are missing due to clerical errors, it is often reasonable to ignore this fact. When the process by which observations are missed are not ignorable, the model (or models) should be enhanced to represent these processes. One simple approach for enhancing a model for $(X_1, \ldots, X_n)$ is to add variables $(I_1, \ldots, I_n)$, where I_i is a binary variable that indicates whether or not the observation of variable X_i in the original model is missing. Rubin (1976) discusses the concept of ignorability and methods for treating non-ignorable data collection processes. The methods for handling missing data that we discuss here assume that the models under consideration have appropriately represented the data collection process.

3.1. The Laplace approximation and related methods

In this subsection and the two that follow, we consider large-sample approximations. The accuracy of some of these approximations depend on the coordinate system used to represent the parameters. In the previous section, where we examined Bayesian networks for discrete variables, we introduced the coordinate system Θ_m corresponding to the parameters $\boldsymbol{\theta}_m$. An alternative coordinate system, which we denote by Φ_m, corresponds to the parameters

$$\phi_{ijk} = \log \frac{\theta_{ijk}}{\theta_{ij1}}$$

for $i = 1, \ldots, n, j = 1, \ldots, q_i, k = 2, \ldots, r_i$. This set of parameters (for fixed i and j) is known as the *natural parameter set* for the multinomial distribution (e.g., Bernardo & Smith, 1994, pp. 199–202). Although Θ_m and Φ_m are equivalent in that there is a one-to-one mapping between them, MacKay (1996) has shown that the use of the natural parameters typically leads to a more accurate approximation of the kind that we consider. Consequently, we use this coordinate system for our approximations. We also use Φ_m for most of our discussions, although sometimes it will be more convenient to express our procedures in terms of Θ_m.

The basic idea behind large-sample approximations is that, as the sample size N increases, $p(\boldsymbol{\phi}_m|D,\mathbf{m}) \propto p(D|\boldsymbol{\phi}_m,\mathbf{m}) \cdot p(\boldsymbol{\phi}_m|\mathbf{m})$ can be approximated as a multivariate-Gaussian distribution. In particular, let

$$g(\boldsymbol{\phi}_m) \equiv \log(p(D|\boldsymbol{\phi}_m,\mathbf{m}) \cdot p(\boldsymbol{\phi}_m|\mathbf{m})). \tag{12}$$

Also, define $\tilde{\boldsymbol{\phi}}_m$ to be the configuration of $\boldsymbol{\phi}_m$ that maximizes $g(\boldsymbol{\phi}_m)$. This configuration also maximizes $p(\boldsymbol{\phi}_m|D,\mathbf{m})$, and is known as the *maximum a posteriori* (MAP) configuration of $\boldsymbol{\phi}_m$ given D. Using a second degree Taylor polynomial of $g(\boldsymbol{\phi}_m)$ about $\tilde{\boldsymbol{\phi}}_m$ to approximate $g(\boldsymbol{\phi}_m)$, we obtain

$$g(\boldsymbol{\phi}_m) \approx g(\tilde{\boldsymbol{\phi}}_m) - \frac{1}{2}(\boldsymbol{\phi}_m - \tilde{\boldsymbol{\phi}}_m)A(\boldsymbol{\phi}_m - \tilde{\boldsymbol{\phi}}_m)^t, \tag{13}$$

where $(\boldsymbol{\phi}_m - \tilde{\boldsymbol{\phi}}_m)^t$ is the transpose of row vector $(\boldsymbol{\phi}_m - \tilde{\boldsymbol{\phi}}_m)$, and A is the negative Hessian of $g(\boldsymbol{\phi}_m)$ evaluated at $\tilde{\boldsymbol{\phi}}_m$. Raising $g(\boldsymbol{\phi}_m)$ to the power of e and using Equation 12, we obtain

$$\begin{aligned} &p(D|\boldsymbol{\phi}_m,\mathbf{m})\; p(\boldsymbol{\phi}_m|\mathbf{m}) \\ &\quad \approx p(D|\tilde{\boldsymbol{\phi}}_m,\mathbf{m})\; p(\tilde{\boldsymbol{\phi}}_m|\mathbf{m})\; \exp\{-\frac{1}{2}(\boldsymbol{\phi}_m - \tilde{\boldsymbol{\phi}}_m)A(\boldsymbol{\phi}_m - \tilde{\boldsymbol{\phi}}_m)^t\}. \end{aligned} \tag{14}$$

Hence, the approximation for $p(\boldsymbol{\phi}_m|D,\mathbf{m}) \propto p(D|\boldsymbol{\phi}_m,\mathbf{m})\; p(\boldsymbol{\phi}_m|\mathbf{m})$ is Gaussian. Integrating both sides of Equation 14 over $\boldsymbol{\phi}_m$ and taking the logarithm, we obtain the approximation

$$\log p(D|\mathbf{m}) \approx \log p(D|\tilde{\boldsymbol{\phi}}_m,\mathbf{m}) \; + \log p(\tilde{\boldsymbol{\phi}}_m|\mathbf{m}) \; + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|A|, \tag{15}$$

where d is the dimension of $\mathbf{m}$—that is, the number of parameters in $\boldsymbol{\phi}_m$. For Bayesian networks satisfying the assumptions described in the previous section, $d = \prod_{i=1}^{n} q_i(r_i - 1)$. This approximation technique for integration is known as *Laplace's method*, and we refer to Equation 15 as the *Laplace approximation*. Kass et al. (1988) have shown that, under certain conditions, the relative error of this approximation, given by

$$\frac{[p(D|\mathbf{m})]_{\text{Laplace}} - [p(D|\mathbf{m})]_{\text{correct}}}{[p(D|\mathbf{m})]_{\text{correct}}},$$

is $O_p(1/N)$, where N is the number of cases in D. Thus, the Laplace approximation can be extremely accurate.

Several of the conditions used by Kass et al. to characterize the accuracy of the Laplace approximation are worth noting, because they are violated in some of our experiments. One condition is that the MAP configuration $\tilde{\boldsymbol{\phi}}_m$ does not lie on the boundary of $\boldsymbol{\phi}_m$. In Section 4.5, we examine how violations of this condition affect the accuracy of the Laplace (and other) approximations.

Another condition used by Kass et al. is that, given D, there is a unique MAP configuration $\tilde{\boldsymbol{\phi}}_m$. When this condition holds, the parameters of the model are said to be *identified*.

There are two common situations in which the parameters of a Bayesian network with hidden variables are not identified. In one case, known as *aliasing*, the state labels of a hidden variable can be interchanged without affecting $p(\boldsymbol{\phi}_m|D,\mathbf{m})$. In Section 3.5, we discuss methods for recovering an accurate Laplace approximation when aliasing occurs. In the other case, the likelihoods $p(\mathbf{x}|\boldsymbol{\phi}_m,\mathbf{m})$ for all $\mathbf{x}$ can be encoded by a smaller set of parameters than $\boldsymbol{\phi}_m$. That is, the dimension of the model is less than the number of parameters in $\boldsymbol{\phi}_m$ (e.g., Geiger, Heckerman, & Meek, 1996). Consequently, the model will have an (uncountably) infinite number of MAP configurations for $\boldsymbol{\phi}_m$. We know of no formal construction of a Laplace approximation that is accurate in this circumstance. Nonetheless, for our experiments, the issue of reduced dimensionality is likely to be mute. In particular, Geiger et al. (1996) provide evidence that the models we examine in this paper do not have redundant parameters.

To compute the Laplace approximation, we must determine $\tilde{\boldsymbol{\phi}}_m$ and the Hessian of $-g(\boldsymbol{\phi}_m)$ evaluated at $\tilde{\boldsymbol{\phi}}_m$. We discuss methods for finding $\tilde{\boldsymbol{\phi}}_m$ in Section 3.2. Meng and Rubin (1991) describe a numerical technique for computing the second derivatives in the Hessian. Raftery (1995) shows how to approximate the Hessian using likelihood-ratio tests that are available in many statistical packages. Thiesson (1997) demonstrates that, for multinomial distributions, the second derivatives can be obtained using Bayesian-network inference. We use Thiesson's method in our experiments.

Although Laplace's approximation is efficient relative to Monte-Carlo approaches, the computation of $|A|$ is nevertheless intensive for large-dimension models. One simplification is to approximate the Hessian A with a block-diagonal matrix, where the entries corresponding to $-\partial^2 g(\boldsymbol{\phi}_m)/\partial\phi_{ijk}\partial\phi_{abc}$ are set to zero, for $i \neq a$. A further simplification is to approximate A using only its diagonal elements. These *Block* and *Diagonal* approximations have been considered by Buntine (1994) and Becker and LeCun (1989), respectively, in feed-forward neural networks. Roughly speaking, in using these approximations, we are forcing independence among parameters that may not in fact be independent.

We obtain another efficient (but less accurate) approximation by retaining only those terms in Equation 15 that increase with N: $\log p(D|\tilde{\boldsymbol{\phi}}_m,\mathbf{m})$, which increases linearly with N, and $\log|A|$, which increases as $d\log N$. Also, for large N, $\tilde{\boldsymbol{\phi}}_m$ can be approximated by $\hat{\boldsymbol{\phi}}_m$, the configuration of $\boldsymbol{\phi}_m$ that maximizes $p(D|\boldsymbol{\phi}_m,\mathbf{m})$, known as the maximum likelihood (ML) configuration of $\boldsymbol{\phi}_m$. Thus, we obtain

$$\log p(D|\mathbf{m}) \approx \log p(D|\hat{\boldsymbol{\phi}}_m,\mathbf{m}) \; - \frac{d}{2}\log N. \tag{16}$$

This approximation is called the *Bayesian information criterion* (BIC). Schwarz (1978) has shown that the relative error of this approximation is $O_p(1)$ for a limited class of models. Haughton (1988) has extended this result to curved exponential models. Kass and Wasserman (1995) and Raftery (1995) have shown that, for particular priors, the BIC has a relative error of $O_p(N^{-1/2})$.

The BIC approximation is interesting in several respects. First, it depends neither on the prior[3] nor the coordinate system of the parameters. Second, the approximation is quite intuitive. Namely, it contains a term measuring how well the parameterized model predicts the data ($\log p(D|\hat{\boldsymbol{\phi}}_m,\mathbf{m})$) and a term that penalizes the complexity of the model

($d/2 \log N$). Third, the BIC approximation is exactly minus the Minimum Description Length (MDL) criterion described by Rissanen (1987).

3.2. Computation of MAP and ML configurations

To compute any of the approximations that we have described, we need to determine either the maximum a posteriori or maximum likelihood configuration for $\boldsymbol{\phi}_m$.

One class of techniques for finding a MAP or ML configuration is gradient-based optimization. For example, we can use gradient ascent, where we follow the derivatives of $p(\boldsymbol{\phi}_m|D, \mathbf{m})$ or $p(D|\boldsymbol{\phi}_m, \mathbf{m})$ to a local maximum. Russell, Binder, Koller, and Kanazawa (1995) and Thiesson (1997) show how to compute the derivatives of the likelihood for a Bayesian network with multinomial distributions. Buntine (1994) discusses the more general case where the local distribution functions come from the exponential family.

Another technique for finding a local MAP or ML configuration is the expectation–maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977). To find a local MAP or ML configuration, we begin by assigning a configuration to $\boldsymbol{\phi}_m$ somehow (e.g., at random). Next, we compute the *expected* sufficient statistics for a complete data set, where expectation is taken with respect to the joint distribution for $\mathbf{X}$ conditioned on the assigned configuration of $\boldsymbol{\phi}_m$ and the known data D. For Bayesian networks with discrete variables, we compute

$$\mathrm{E}_{p(\mathbf{x}|D,\phi_m,\mathbf{m})}(N_{ijk}) = \sum_{l=1}^{N} p(x_i^k, \mathbf{pa}_i^j|\mathbf{x}_l, \boldsymbol{\phi}_m, \mathbf{m}), \tag{17}$$

where $\mathbf{x}_l$ is the possibly incomplete lth case in D. When X_i and all the variables in $\mathbf{Pa}_i$ are observed in case $\mathbf{x}_l$, the corresponding term for this case requires a trivial computation: it is either zero or one. Otherwise, we can use any Bayesian-network inference algorithm (e.g., Jensen, Lauritzen, & Olesen, 1990) to evaluate the term. This computation is called the *expectation step* of the EM algorithm.

Next, we use the expected sufficient statistics as if they were actual sufficient statistics from a complete random sample D_c. If we are doing an ML calculation, then we determine the configuration of $\boldsymbol{\phi}_m$ that maximizes $p(D_c|\boldsymbol{\phi}_m, \mathbf{m})$. This configuration is given by $\phi_{ijk} = \log(\theta_{ijk}/\theta_{1jk})$, where

$$\theta_{ijk} = \frac{\mathrm{E}_{p(\mathbf{x}|D,\phi_m,\mathbf{m})}(N_{ijk})}{\sum_{k=1}^{r_i} \mathrm{E}_{p(\mathbf{x}|D,\phi_m,\mathbf{m})}(N_{ijk})}.$$

If we are doing a MAP calculation, then we determine the configuration of $\boldsymbol{\phi}_m$ that maximizes the posterior density of the parameters. When working with the coordinate system Φ_m, this configuration is given by $\phi_{ijk} = \log(\theta_{ijk}/\theta_{1jk})$, where

$$\theta_{ijk} = \frac{\alpha_{ijk} + \mathrm{E}_{p(\mathbf{x}|D,\phi_m,\mathbf{m})}(N_{ijk})}{\sum_{k=1}^{r_i}(\alpha_{ijk} + \mathrm{E}_{p(\mathbf{x}|D,\phi_m,\mathbf{m})}(N_{ijk}))}.$$

This assignment is called the *maximization step* of the EM algorithm. Dempster et al. (1977) showed that iteration of the expectation and maximization steps will converge to a

local maximum. The EM algorithm is typically applied when sufficient statistics exist (i.e., when local distribution functions are in the exponential family), although generalizations of the EM algorithm have been used for more complicated local distributions (e.g., Saul, Jaakkola, & Jordan, 1996).

The models that we consider often have more than one local maximum. Consequently, these techniques will not necessarily find the global MAP or ML configuration. One often-used partial solution to this problem is to start from many (usually random) initial configurations of $\boldsymbol{\phi}_m$. We use a variant of this technique in our experiments.

3.3. The Cheeseman–Stutz approximation

Another approximation for the marginal likelihood is based on the fact that $p(D|\mathbf{m})$ can be computed efficiently for complete data. Consider the equality

$$p(D|\mathbf{m}) = p(D'|\mathbf{m}) \frac{\int p(D, \boldsymbol{\phi}_m|\mathbf{m})\, d\boldsymbol{\phi}_m}{\int p(D', \boldsymbol{\phi}_m|\mathbf{m})\, d\boldsymbol{\phi}_m}, \tag{18}$$

where D' is any completion of the data set D. Because D' is a complete data set, we can compute $p(D'|\mathbf{m})$ using Equation 11. Now, suppose we apply Laplace approximations to the numerator and denominator of the second term. Roughly speaking, the resulting approximation for $p(D)$ will be best if the quantities $p(D, \boldsymbol{\phi}_m|\mathbf{m})$ and $p(D', \boldsymbol{\phi}_m|\mathbf{m})$—regarded as functions of $\boldsymbol{\phi}_m$—are similar in shape, so that errors in the two Laplace approximations tend to cancel. The two functions cannot be similar in an absolute sense, because D' contains more information than does D, and hence $p(D', \boldsymbol{\phi}_m|\mathbf{m})$ will be more peaked than $p(D, \boldsymbol{\phi}_m|\mathbf{m})$. Nonetheless, we can make the two functions more similar by completing D' so that they peak for the same configuration of $\boldsymbol{\phi}_m$. That is, we want $\tilde{\boldsymbol{\phi}}'_m$, the MAP configuration of $\boldsymbol{\phi}_m$ given D', to be equal to $\tilde{\boldsymbol{\phi}}_m$. One way to obtain this equality is to complete D' so that its sufficient statistics match the expected sufficient statistics given D and $\mathbf{m}$. In the case of Bayesian networks with discrete variables and multinomial distributions, this completion is given by

$$N'_{ijk} = \mathrm{E}_{p(\mathbf{x}|D, \phi_m, \mathbf{m})}(N_{ijk}) \tag{19}$$

for all i, j, and k, where the N'_{ijk} are the sufficient statistics for D'. This choice for D' is also desirable from a computational standpoint, because—when using the EM algorithm to find $\tilde{\boldsymbol{\phi}}_m$—the sufficient statistics N'_{ijk} are computed in the last expectation step.

Applying the Laplace approximation Equation 15 to the numerator and denominator of Equation 18, and using the fact that $\tilde{\boldsymbol{\phi}}'_m = \tilde{\boldsymbol{\phi}}_m$, we obtain

$$\log \mathrm{p}(\mathrm{D}|\mathbf{m}) \approx \log \mathrm{p}(\mathrm{D}'|\mathbf{m}) - \log \mathrm{p}(\mathrm{D}'|\tilde{\phi}_\mathrm{m}, \mathbf{m}) + \frac{1}{2} \log |\mathrm{A}'| + \log \mathrm{p}(\mathrm{D}|\tilde{\phi}_\mathrm{m}, \mathbf{m}) - \frac{1}{2} \log |\mathrm{A}|, \tag{20}$$

where A' is the negative Hessian of $\log p(D', \boldsymbol{\phi}_m|\mathbf{m})$ evaluated at $\tilde{\boldsymbol{\phi}}_m$. Because we derive this approximation using two Laplace approximations, Equation 20 must have a relative

error that is no worse than $O_p(1/N)$. Nonetheless, a careful derivation may show that it is more accurate.

A more efficient approximation is obtained by applying the BIC/MDL approximation to the numerator and denominator of Equation 18. In this case, we have

$$\log \mathrm{p}(\mathrm{D}|\mathbf{m}) \approx \log \mathrm{p}(\mathrm{D}'|\mathbf{m}) - \log \mathrm{p}(\mathrm{D}'|\tilde{\phi}_{\mathrm{m}}, \mathbf{m}) + \frac{\mathrm{d}'}{2} \log \mathrm{N} + \log \mathrm{p}(\mathrm{D}|\tilde{\phi}_{\mathrm{m}}, \mathbf{m}) - \frac{\mathrm{d}}{2} \log \mathrm{N} \quad (21)$$

where we have used the MAP rather than ML configuration for $\boldsymbol{\phi}_m$, and we have allowed for the possibility that d', the dimension of $\mathbf{m}$ for complete data, may be greater than the dimension of $\mathbf{m}$ for the actual data. Equation 21 (without the correction for dimension) was introduced by Cheeseman and Stutz (1995) for use as a model-selection criterion in AutoClass. We shall refer to Equation 21 as the *Cheeseman–Stutz* approximation. We note that this approximation can be easily extended to any statistical model that has sufficient statistics. For example, the Cheeseman–Stutz approximation can be applied to any Bayesian network with local distribution functions from the exponential family. Our heuristic derivation of the Cheeseman–Stutz approximation does not tell us whether it is better to use the MAP or ML configuration in the approximation. Thus, we examine both alternatives in our experiments.

3.4. Monte-Carlo methods

We now discuss Monte-Carlo methods, concentrating on the method we use to evaluate the accuracy of the large-sample approximations.

A common Monte-Carlo method, introduced by Geman and Geman (1984), is known as *Gibbs sampling*. Given variables $\mathbf{X} = (X_1, \ldots, X_n)$ with some joint distribution $p(\mathbf{x})$, we can use a Gibbs sampler to approximate the expectation of a function $f(\mathbf{x})$ with respect to $p(\mathbf{x})$ as follows. First, we choose an initial state for each of the variables in $\mathbf{X}$, say at random. Next, we unassign the current state of X_1 and compute its probability distribution given the configuration of the other $n - 1$ variables. We repeat this procedure for each variable $X_2, \ldots, X_n$, thus creating a new sample of $\mathbf{x}$. We then iterate the previous steps, keeping track of the simple average of $f(\mathbf{x})$ over the samples we construct. After a (usually small) number of iterations—known as the "burn-in" phase—the possible configurations of $\mathbf{x}$ will be sampled with probability $p(\mathbf{x})$.[4] Consequently, the simple average of $f(\mathbf{x})$ over these samples will converge to $\mathrm{E}_{p(\mathbf{x})}(f(\mathbf{x}))$. Introductions to Gibbs sampling and other Monte-Carlo methods—including discussions of convergence—are given by Neal (1993) and by Madigan and York (1995).

The particular approach we use to compute the marginal likelihood is known as the *Candidate method* (Chib, 1995; Raftery, 1996). The approach is based on Bayes' theorem, which says that

$$p(D|\mathbf{m}) = \frac{p(D|\boldsymbol{\phi}^*_m, \mathbf{m})\; p(\boldsymbol{\phi}^*_m|\mathbf{m})}{p(\boldsymbol{\phi}^*_m|D, \mathbf{m})}$$

for any configuration $\boldsymbol{\phi}^*_m$. To compute $p(D|\mathbf{m})$, we choose some configuration $\boldsymbol{\phi}^*_m$, evaluate the numerator exactly, and approximate the denominator using a Gibbs sampler.

To approximate $p(\boldsymbol{\phi}^*_m|D,\mathbf{m})$, we first initialize the states of the unobserved variables in each case. As a result, we have a complete random sample D_c. Second, we choose some variable X_{il} (variable X_i in case l) that is not observed in the original random sample D, and reassign its state according to the probability distribution

$$p(x'_{il}|D_c \setminus x_{il},\mathbf{m}) = \frac{p(x'_{il}, D_c \setminus x_{il}|\mathbf{m})}{\sum_{x''_{il}} p(x''_{il}, D_c \setminus x_{il}|\mathbf{m})},$$

where $D_c \setminus x_{il}$ denotes the data set D_c with observation x_{il} removed, and the sum in the denominator runs over all states of variable X_{il}. The terms in the numerator and denominator are marginal likelihoods for complete data, and thus can be computed using Equation 11. Third, we repeat this reassignment for all unobserved variables in D, producing a new complete random sample D'_c. Fourth, we compute the posterior density $p(\boldsymbol{\phi}^*_m|D'_c,\mathbf{m})$ using Equations 9 and 10 (adjusted for the coordinate system Φ_m). Finally, we iterate the previous three steps, and use the simple average of $p(\boldsymbol{\phi}^*_m|D'_c,\mathbf{m})$ as our approximation.

In principle, the Candidate method can be applied using any configuration $\boldsymbol{\phi}^*_m$. Nonetheless, certain configurations lead to faster convergence of the Gibbs sampler. Chib (1995) and Raftery (1996) suggest that $\tilde{\boldsymbol{\phi}}_m$ be used. Nonetheless, in experiments with multinomial-mixture models, we have found that the use of this configuration underestimates $p(\boldsymbol{\phi}^*_m|D,\mathbf{m})$. This error occurs because, when $\tilde{\boldsymbol{\phi}}_m$ is used, there are configurations of D_c such that (1) $p(\boldsymbol{\phi}^*_m|D_c,\mathbf{m})$ is extremely large, and (2) the configuration D_c is extremely unlikely to be visited. Consequently, when these configurations of D_c are not visited in a particular run, the simple average of $p(\boldsymbol{\phi}^*_m|D_c,\mathbf{m})$ is substantially less than $p(\boldsymbol{\phi}^*_m|D,\mathbf{m})$.

We have experimented with an alternative method for choosing $\boldsymbol{\phi}^*_m$. For a fixed number of samples after the burn-in phase, we keep track of the configurations of D_c. After these samples have been collected, we retain the configuration D^*_c that occurred most frequently. We break ties by choosing the configuration with the largest value of $p(D_c|\mathbf{m})$. Finally, we set $\boldsymbol{\phi}^*_m$ to be the configuration that maximizes $p(\boldsymbol{\phi}_m|D^*_c,\mathbf{m})$. In experiments with multinomial-mixture models, such as those presented in Section 4.4, this choice of $\boldsymbol{\phi}^*_m$ yields low-noise estimates of $p(D|\mathbf{m})$.

3.5. Hidden-variable models and aliasing

Given a Bayesian network $\mathbf{m}$ for $\mathbf{X}$, suppose $X_i \in \mathbf{X}$ is never observed in data set D. Because X_i is hidden, the likelihood $p(D|\boldsymbol{\phi}_m,\mathbf{m})$ will be invariant to arbitrary relabelings of the states of X_i. Thus, if the prior $p(\boldsymbol{\phi}_m|\mathbf{m})$ is invariant to such relabelings, so will be the posterior $p(\boldsymbol{\phi}_m|D,\mathbf{m})$. It follows that if $\tilde{\boldsymbol{\phi}}_m$ is a MAP configuration of $\boldsymbol{\phi}_m$, then there will be additional MAP configurations corresponding to the relabelings of the states of X_i. We shall refer to each such configuration and its neighborhood as an *alias*. If each alias is distinct (i.e., nondegenerate), then there will be $r_i!$ of them.

When there are multiple distinct aliases, the parameters of $\mathbf{m}$ are no longer identifiable. Nonetheless, assuming the aliases are well separated, we can apply the Laplace approximation locally around each of them, summing the contributions of each peak. Assuming one hidden variable and distinct aliases, this procedure amounts to multiplying the marginal

likelihood corresponding to one alias by $r_i!$. This correction applies to the Block, Diagonal, BIC/MDL, and Cheeseman–Stutz approximations as well.

With sufficient computation, the Candidate approximation for $p(D|\mathbf{m})$ does not need to be corrected for aliases, because the Gibbs sampler will visit all assignments to the hidden variable(s). In our experiments, however, the Gibbs sampler tends to stay near one alias. We can compensate for this failure to move among aliases by multiplying the approximation for marginal likelihood by $r_i!$, as we do for the large-sample approximations. We obtain a more accurate correction, however, by (in effect) running $r_i!$ Gibbs samplers in parallel. In particular, for every completion D_c that we actually visit, we compute $p(\boldsymbol{\phi}_m^*|D_c', \mathbf{m})$ for each equivalent assignment D_c', and average the results. To compute $p(\boldsymbol{\phi}_m^*|D, \mathbf{m})$, we then average these averages.[5] This procedure yields an accurate correction factor even when the Gibbs sampler moves among aliases and when there are degenerate aliases. The procedure is expensive for large r_i, but was not prohibitive for our experiments.

3.6. Computational complexity

The accuracy of these approximations should be balanced against their computation costs. These costs will depend on the topology of the Bayesian network under consideration. Here, we consider costs for an arbitrary Bayesian network with discrete variables and a naive-Bayes discrete-variable clustering model of the form shown in Figure 1 (a multinomial-mixture model). In both cases, we assume that the EM algorithm is used to find a MAP or ML configuration of $\boldsymbol{\phi}_m$.

For an arbitrary Bayesian network, the evaluation of Cheeseman–Stutz, Diagonal, and BIC/MDL is dominated by the determination of the MAP or ML configuration of the parameters. The time complexity of this task is $O(eiN + ed)$, where e is the number of EM iterations and i is the cost of inference in Equation 17.[6]

The evaluation of the Laplace approximation typically is dominated by the computation of the Hessian determinant. The time complexity of this computation (using Thiesson's 1997 method) is $O(diN + d^3)$. Because $i > d$ and (typically) $N > d$, the computation of the Hessian determinant is $O(diN)$. The Block approximation has the same complexity as the Laplace approximation, because one block may contain most of the parameters.

For the naive-Bayes clustering model, the evaluation of the Cheeseman–Stutz, Diagonal, and BIC/MDL measures are again dominated by the determination of the MAP or ML configuration. In the expectation step, we compute—for each case—the posterior probability of the hidden variable given the observed variables and the parameters. Thus, the cost of MAP/ML determination is $O(edN)$.

The Laplace approximation is again dominated by the computation of the Hessian determinant, having a cost of $O(d^2N)$. The computational cost of the Block approximation has two components. The cost of the MAP/ML determination is $O(edN)$. The Hessian contains $O(n)$ blocks each of size $O(r_c)$, where r_c is the number of states of the hidden variable; consequently, the evaluation of the Hessian costs $O(r_c^2 nN) = O(r_c dN)$. Thus, the overall cost of the Block computation is $O(r_c dN + enN)$.

4. Experiments with multinomial-mixture models

Our primary goal is to evaluate the accuracy and efficiency of the Block, Diagonal, BIC/MDL, and Cheeseman–Stutz approximations when used for model averaging and model selection among hidden-variable Bayesian networks. We evaluate the Cheeseman–Stutz approximation using both the maximum a posteriori (CS MAP) and maximum likelihood (CS ML) configurations of $\boldsymbol{\phi}_m$. Similarly, we evaluate the BIC/MDL approximation using both MAP and ML configurations. A secondary goal is to evaluate the accuracy of the Laplace approximation when applied to hidden-variable Bayesian networks.

Our approach is straightforward. For a variety of models and data sets, we compare values for the marginal likelihood given by the various approximations with that given by a Monte-Carlo method that we believe to be accurate. In addition, we measure the time required to compute each approximation.

The models we evaluate are the naive-Bayes clustering models of the form shown in Figure 1. We consider synthetic models and data sets as well as models for real-world data sets. For a particular data set, we compute approximate marginal likelihoods for a series of naive-Bayes *test models*, where the only difference among test models is the number of states r_c of the hidden variable C. We begin with a test model with $r_c = 1$, which corresponds to a model where the observed variables $(X_1, \ldots, X_n)$ are mutually independent. We then increase r_c, typically observing a peak in the marginal likelihood, until the marginal likelihood as determined by all approximations is clearly decreasing. To evaluate the accuracy of an approximation for the purpose of model selection, we compare the value of r_c that would be selected using that approximation with the value of r_c that would be selected using the Monte-Carlo standard. To evaluate the accuracy of an approximation for the purpose of model averaging, we examine how each approximation weighs the second most likely model relative to the most likely model.

In our evaluations of data sets generated from synthetic models, the true number of states of the hidden variable (r_{ct}) is available. Nonetheless, we do not use these values in our evaluation of the approximations. In particular, we are interested in how well the various methods approximate the marginal likelihood. A comparison between the best value for r_c selected by an approximation and r_{ct} would only serve to introduce confounding factors in this evaluation. For example, although the true model may have $r_{ct} = 4$, the sample size of the data may not be sufficiently large to support a mixture model with four components. Nonetheless, an approximation that tends to select models that are too large may (by chance) select $r_c = 4$. Consequently, if we were to use $r_{ct} = 4$ for our comparison, we would incorrectly deem this selection to be a success.

4.1. Experimental parameters

All experiments were run on a P6 200MHz machine under the Windows NT^{TM} operating system. The various algorithms were implemented in C++.

We used the method of Thiesson (1997) to evaluate the Hessian of $-\log p(\boldsymbol{\phi}_m, D|\mathbf{m})$. To compute the Cheeseman–Stutz scoring function, we assumed that dimensions d' and d

were equal. Although we know of no proof that this assumption is correct, Geiger et al. (1996) provide evidence that the relation holds.

We used the EM algorithm to determine the MAP and ML configurations needed by the approximations. We determined MAP configurations in the coordinate system Φ_m. The EM algorithm ran until either the relative difference between successive values for $p(\boldsymbol{\phi}_m|D,\mathbf{m})$ (or $p(D|\boldsymbol{\phi}_m,\mathbf{m})$) was less than 10^{-8} or 400 iterations were reached. In preliminary experiments, substantial additional iterations led to relative differences in the approximations of less than 10^{-4}. Such differences did not have a significant effect on results.

In order to avoid local MAP and ML configurations, we used a variant of the multiple-restart approach described in Section 3.2. First, we sampled 64 configurations of the parameters $\boldsymbol{\phi}_m$ from distributions that are uniform in Θ_m. Next we performed one expectation and maximization step, and retained the 32 initial configurations that led to the largest values of $p(\boldsymbol{\phi}_m|D,\mathbf{m})$. Then we performed two expectation and maximization steps, retaining the 16 best initial configurations. We continued this procedure, doubling the number of expectation-maximization steps at each iteration, until only one configuration remained.

The exact marginal likelihood for test models with a single mixture component ($r_c = 1$) can be computed in closed form (Equation 11). We used these exact values in lieu of approximate values for all experiments. In Section 4.4, we discuss the parameters of the Monte-Carlo standard.

In all experiments, we used a uniform prior distribution over model structures, $p(\mathbf{m}) =$ constant. Consequently, $p(\mathbf{m}|D) \propto p(D|\mathbf{m})$, and the value r_c selected by a particular approximation method corresponded to the value of r_c for which that method's marginal likelihood was a maximum. We denote this value by r_{c*}. We used Dirichlet prior distributions given by $\alpha_{ijk} = 1$ (uniform in Θ_m) in all experiments except the one in which we investigated sensitivity to parameter priors.

4.2. Preliminary experiment

The goal of our first experiment was to gain a rough understanding of the accuracy of the approximations. We performed this study with synthetic models and data. In particular, we generated naive-Bayes models for various values of n and r_{ct}. For each n and r_{ct} considered, we created a model in which each observed variable X_i had two states. For each model, we set the parameters of the root node to be uniform (in Θ_m), and sampled the remaining parameters from a uniform distribution (in Θ_m). We then generated data with sample size N from the model using a forward sampling technique. That is, we sampled a state $C = c^j$ according to $p(C)$, and then sampled a state of each X_i according to $p(x_i|C = c^j)$. Finally, we discarded the samples of C, retaining only the samples of $X_1, \ldots, X_n$.

For values that we considered—$n = 32, 64, 128$, $r_{ct} = 4, 6, 8$, and $N = 50, 100, 200, 400$—we obtained plots of $\log p(D|\mathbf{m})$ versus r_c that were similar in form. A typical plot for $n = 64$, $r_c = 4$, and $N = 400$ is shown in Figure 2(a). Overall, the Candidate, Laplace, Block, Diagonal, and Cheeseman–Stutz MAP approximations usually peaked at the same

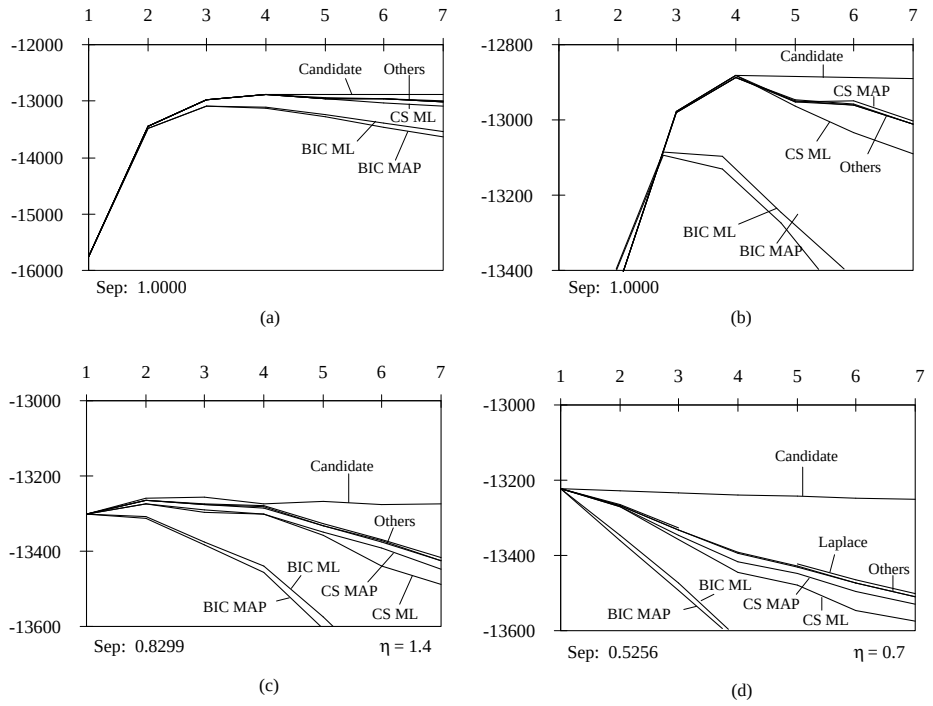

Figure 2. Plots of $\log p(D|\mathbf{m})$ versus r_c for synthetic data sets with $n = 64$, $r_{ct} = 4$, and $N = 400$. (a) The baseline wherein each variable x_i has two states and model parameters are sampled from independent distributions. (b) A higher resolution view of (a). (c) Model parameters are dependent with $\eta = 1.4$. (d) Model parameters are dependent with $\eta = 0.7$. The separation scores for each experimental condition are shown below its plot. "CS" is an abbreviation for "Cheeseman–Stutz".

value of r_c. The Laplace, Block, Diagonal, and Cheeseman–Stutz MAP approximations usually agreed with the Monte-Carlo standard for $r_c \leq r_{c*}$, but fell below the standard for $r_c > r_{c*}$. The BIC/MDL approximation peaked for smaller values of r_c and decreased more sharply to the right of the peak than did the other approximations. The Cheeseman–Stutz approximation was more accurate when the MAP configuration was used, whereas the BIC/MDL approximation was more accurate when the ML configuration was used.

These experiments were informative, but they did not help to discriminate the Laplace, Block, Diagonal, and Cheeseman–Stutz approximations. After additional experiments, we identified a likely cause: the clusters were well separated. In Section 4.3, we examine this phenomenon and describe more challenging data sets for analysis.

Before we do so, consider the observation that the large-sample approximations yield values that fall below those of the Monte-Carlo standard for $r_c > r_{c*}$. One possible explanation is that many local MAP configurations may exist when $r_c > r_{c*}$. If this

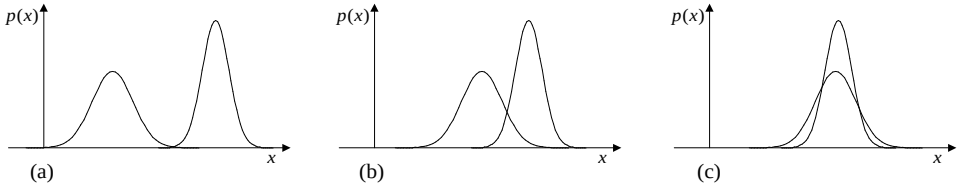

Figure 3. Gaussian mixtures in which the two components are (a) well separated, (b) partially separated, and (c) poorly separated.

condition occurs, then the marginal likelihood could be significantly underestimated by a Laplace approximation around a single local maximum. To test this hypothesis, we used random restarts in several of our experiments to visit hundreds of different local maxima, summing the contributions to the marginal likelihood from each maximum. In no case, however, did this approach improve performance significantly.

Another explanation is that, when $r_c > r_{c*}$, the test model will contain more classes than are needed to fit the data. Thus, it is likely that some of the classes will be *empty* in the sense that $p(C = c^j|\tilde{\boldsymbol{\phi}}_m, \mathbf{m})$ will be close to zero for some c^j, and the parameters corresponding to the conditional probabilities of the empty classes will be superfluous. As a result, the posterior distribution $p(\boldsymbol{\phi}_m|D, \mathbf{m})$ will be a ridge rather than a peak, and the large-sample approximations, which assume the posterior distribution is a peak, will underestimate the marginal likelihood. In almost all of our experiments, we have found that some of the classes are empty when $r_c > r_{c*}$.

4.3. Cluster separation

The concept of cluster separation is difficult to visualize for multinomial-mixture models. To understand this concept, let us consider one-dimensional Gaussian-mixture models as shown in Figure 3. Each model contains two Gaussian components. As we move from left to right, the components become less separated. If the mixtures are well separated, as in Figure 3(a), then for most values of x, $p(c^j|x) = 1$ for either $j = 1$ or $j = 2$. If the mixtures are poorly separated, as in Figure 3(c), then for most values of x, $p(c^j|x) = p(c^j), j = 1, 2$. Generalizing these observations to mixtures of arbitrary distributions, we can think of cluster separation as the degree to which we are certain about the state of C a posteriori, averaged over all possible observations $\mathbf{x}$.

When clusters are well separated in this sense, the learning task is straightforward. In particular, each observation will belong to one class (i.e., one state of C) with high probability. Thus, it is not surprising that the approximations do well. To evaluate the degree of separation of our models, we define the separation score of a model $(\mathbf{m}, \boldsymbol{\phi}_m)$ to be the negative expected entropy of the posterior distribution for C scaled to the range $[0, 1]$:

$$\mathrm{Sep}(\mathbf{m}, \phi_{\mathrm{m}}) = 1 - \frac{1}{\log \mathrm{r_c}} \sum_{\mathbf{x}} \mathrm{p}(\mathbf{x}|\phi_{\mathrm{m}}, \mathbf{m}) \left[\sum_{\mathrm{k}=1}^{\mathrm{r_c}} -\mathrm{p}(\mathrm{c^j}|\mathbf{x}, \phi_{\mathrm{m}}, \mathbf{m}) \log \mathrm{p}(\mathrm{c^j}|\mathbf{x}, \phi_{\mathrm{m}}, \mathbf{m}) \right]. \quad (22)$$

Because the sum over $\mathbf{x}$ is intractable, we use the finite-sample version of Equation 22, which depends on the random sample D:

$$\mathrm{Sep}(\mathbf{m}, \boldsymbol{\phi}_m, D) = 1 - \frac{1}{N \log r_c} \sum_{\mathbf{x} \in D} \sum_{k=1}^{r_c} -p(c^j|\mathbf{x}, \boldsymbol{\phi}_m, \mathbf{m}) \log p(c^j|\mathbf{x}, \boldsymbol{\phi}_m, \mathbf{m}). \quad (23)$$

Note that values for $\mathrm{Sep}(\mathbf{m}, \boldsymbol{\phi}_m, D)$ increase with increasing separation. The separation score for the model in Figure 2(a) is 1.0000, confirming our observation that the clusters are well separated.

To provide the approximations with more of a challenge, we should decrease model separation. One approach for doing so is to decrease n, the number of observed variables. This approach is not useful, however, because we want to evaluate the accuracy and efficiency of the approximations for a wide range of n. Another approach is to sample the parameters from a distribution that is biased toward a uniform distribution (in Θ_m). We do not use this approach either, because we do not believe such parameter distributions are common.

Another approach that produces more realistic models is to introduce dependencies among the parameters such that $p(x_i|c^j, \boldsymbol{\phi}_m, \mathbf{m})$ and $p(x_i|c^l, \boldsymbol{\phi}_m, \mathbf{m})$ are more likely to have similar values for $l \neq j$ than if they were chosen independently. Consider a simple approach for introducing such dependencies, in which we let $\theta(x_i^k|c^j)$ and $\phi(x_i^k|c^j)$ be the parameters corresponding to $p(x_i^k|c^j)$ in the coordinate systems Θ_m and Φ_m, respectively. First, we sample the parameters $\theta(x_i^k|c^1)$ for all i from uniform distributions and transform these parameters to Φ_m. Then, we set

$$\phi(x_i^k|c^j) = \phi(x_i^k|c^1) + \mathrm{Normal}(0, \eta)$$

for $i = 1, \ldots, n$, $j = 2, \ldots, r_c$, and $k = 2, \ldots, r_i$, where $\mathrm{Normal}(0, \eta)$ is a sample from a normal distribution with mean zero and standard deviation η. As η decreases toward zero, the cluster separation decreases.

As shown in Figure 2, when we decrease cluster separation, the value r_{c*} decreases. This observation is not surprising. In the extreme case $\eta = 0$, the clusters are superimposed, and r_{c*} should be one. Given this observation, we want to challenge the approximations with clusters that partially overlap, but not by so much that (in effect) only one cluster remains.

4.4. Monte-Carlo standard

Before we consider additional experiments, let us examine our Monte-Carlo gold standard: the Candidate method.

Recall that the Candidate method uses a Gibbs sampler to determine $p(\boldsymbol{\phi}_m^*|D, \mathbf{m})$ (Section 3.4). This Gibbs sampler has four parameters: α, the number of samples D_c used to burn in the Gibbs sampler; β, the number of samples used to select $\boldsymbol{\phi}_m^*$; γ, the number of

samples that separate the phase where $\boldsymbol{\phi}_m^*$ is selected and the phase where $p(\boldsymbol{\phi}_m^*|D, \mathbf{m})$ is computed; and δ, the number of samples used to compute $p(\boldsymbol{\phi}_m^*|D, \mathbf{m})$. In preliminary experiments with the Candidate method, we increased these parameters until we obtained a low-noise approximation for $\log p(D|\mathbf{m})$ across the spectrum of values $n = 32, 64, 128$, $r_{ct} = 4, 6, 8$, and $N = 50, 100, 200, 400$. We evaluated the noise in the approximation for a given value of the parameters by examining plots of $\log p(D|\mathbf{m})$ versus r_c.

We found that the burn-in and $\boldsymbol{\phi}_m^*$ selection phases could be combined without increasing the noise in the approximation. This observation is not surprising, because typical (and likely) configurations for D_c usually do not occur until the Gibbs sampler has burned in. Except when N was small ($N \leq 50$), no configuration of D_c was visited more than once. Consequently, in most experiments, the configuration D_c^* chosen to select $\boldsymbol{\phi}_m^*$ was the most likely D_c. For $r_{ct} \leq 4$, we found that $\beta = 100, \gamma = 10$, and $\delta = 100$ produced a low-noise approximation. For $r_{ct} \geq 8$, we found that $\beta = 400, \gamma = 10$, and $\delta = 400$ was adequate. Also, the noise in the approximation was slightly lower when we sampled an initial D_c from the MAP configuration of $\boldsymbol{\phi}_m$ rather than from a distribution that is uniform in Θ_m. We used these algorithm parameters in our experiments (including those on real-world data sets).

As we discussed in Section 4.2, the Candidate and Laplace approximations produced similar values for $p(D|\mathbf{m})$ for $r_c \leq r_{c*}$ (see Figure 2), and disagreements for $r_c > r_{c*}$ could be explained. These observations, combined with the fact that the approximation had low noise, suggested that the Candidate approximation was accurate. Nonetheless, the Candidate approximation became noisy for $n < 32$, even when we increased β and δ to 1600. Furthermore, low noise does not guarantee high accuracy. Consequently, we wanted to further evaluate the accuracy of the Candidate method. To do so, we considered data sets with small sample sizes, so that we could determine $p(D|\mathbf{m})$ exactly by summing $p(D_c|\mathbf{m})$ over all possible configurations of D_c consistent with D. We used data generated from synthetic models with $n = 64$, $r_{ct} = 2$, and $N = 10$. Results for various degrees of overlap ($\eta = 1.25, 3.5, 5.75$) are shown in Figure 4.

In all plots, the Candidate approximation agreed closely with the exact value for $p(D|\mathbf{m})$. In addition, although large-sample approximations are unlikely to be valid for samples of size 10, the relationships among the Candidate, Laplace, Block, and Diagonal approximations for $N = 10$ were similar to those for large N. In particular, the Laplace, Block, and Diagonal approximations agreed with the Candidate approximation for $r_c \leq r_{c*}$, but fell below the Candidate approximation for $r_c > r_{c*}$. These results provide additional evidence that the Candidate approximation is accurate.

We note that the Cheeseman–Stutz MAP approximation agreed more closely with the Candidate (and exact) values for $p(D|\mathbf{m})$ than did the Laplace approximation. We suggest an explanation for this observation in the following section.

4.5. Sensitivity analyses for synthetic data

We evaluated the approximations for a variety of synthetic models and data sets. First, we examined the accuracy of the approximations as a function of n (the number of input variables), r_{ct} (the number of classes in the generative model), and N (the sample size

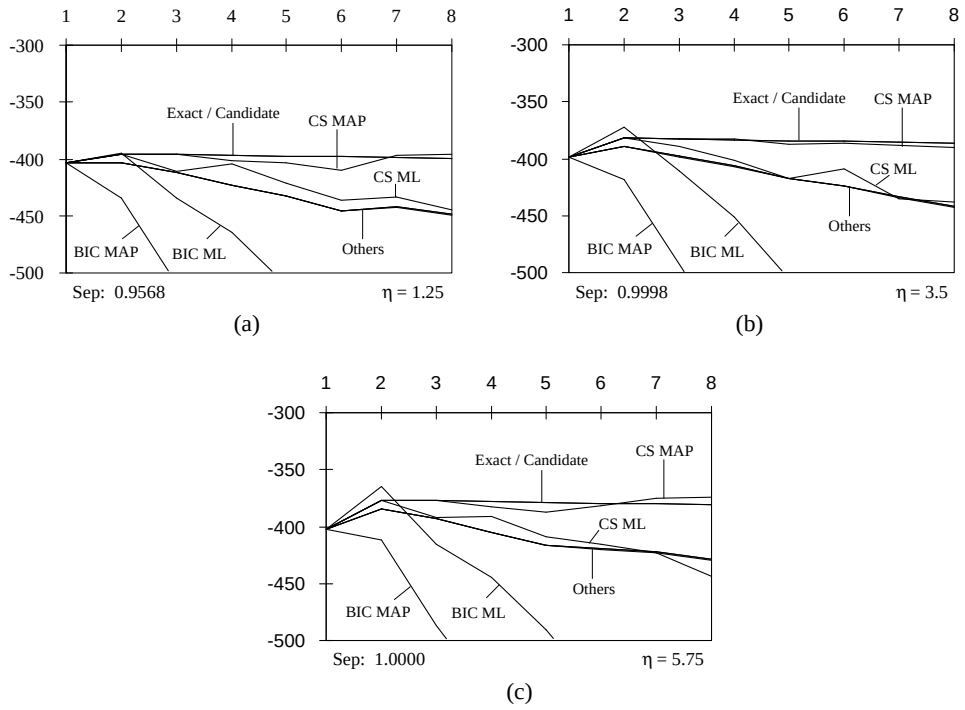

Figure 4. Plots of $\log p(D|\mathbf{m})$ versus r_c for synthetic data sets with $n = 64$, $r_{ct} = 2$, $N = 10$, and various degrees of cluster overlap.

of the data). For each n and r_{ct} considered, we created a model in which each observed variable X_i had two states. For each model, we sampled the parameters for its hidden node from a uniform distribution (in Θ_m) so as to generate clusters of various sizes. We generated dependent parameters for the conditional distributions as described in Section 4.3 using $\eta = 1.75$. For most experiments, this choice for η produced clusters that overlapped partially but not completely. For each experiment—defined by a given n, r_c, and N—we evaluated the approximations for five data sets generated with different random seeds.

Figures 5, 6, and 7 show plots of $\log p(D|\mathbf{m})$ versus r_c for one of the five data sets in the experiments where n, r_c, and N were varied, respectively. The most surprising aspect of the results was that the introduction of cluster overlap did not lead to significant differences in the accuracy of the approximations. As in the case of no cluster overlap, the Candidate, Laplace, Block, Diagonal, and Cheeseman–Stutz MAP approximations usually peaked at the same value of r_c. The Laplace, Block, Diagonal, and Cheeseman–Stutz MAP approximations usually agreed with the Monte-Carlo standard for $r_c \leq r_{c*}$, but fell below the standard for $r_c > r_{c*}$. The BIC/MDL approximation peaked for smaller values of r_c and decreased more sharply to the right of the peak than did the other approximations.

The Cheeseman–Stutz approximation was more accurate when the MAP configuration was used, whereas the BIC/MDL approximation was more accurate when the ML configuration was used.

To evaluate the accuracy of the approximations when used for model selection, we computed the quantity Δr_{c*}—the difference between r_{c*} for the Monte-Carlo standard and r_{c*} for the approximation—and averaged this difference over the five data sets for each experiment. Table 1 contains these averages. With the exception of the Cheeseman–Stutz ML and BIC/MDL approximations, the approximations almost always select the same model.

To evaluate the accuracy of the approximations when used for model averaging, we examined how each approximation penalized the second most likely model relative to the most likely model. In particular, we used the Candidate method to identify the two model structures with the largest ($\mathbf{m}_1$) and second largest ($\mathbf{m}_2$) marginal likelihoods. We then computed the log Bayes factor $\log p(D|\mathbf{m}_1)/p(D|\mathbf{m}_2)$ for each approximation. If an approximation were useful for model averaging, then these scores would be similar to that for the Monte-Carlo standard. The results for the Laplace approximation are shown in Table 2. In 20 of the 40 unique entries, the log Bayes factor was less than $3.6 = \ln(37)$ for the Monte-Carlo standard, but greater than $14.4 = \ln(1.8 \text{ million})$ for the Laplace approximation. In these cases, if we had used the Laplace approximation, we would have removed $\mathbf{m}_2$ from consideration. In contrast, if we had used the Monte-Carlo standard, $\mathbf{m}_2$ would have contributed to the average, perhaps significantly, depending on the hypothesis of interest. Therefore, in (at least) these 20 cases, the Laplace approximation was not a good substitute for the Monte-Carlo standard. The other approximations were at least as inaccurate.

Next, we examined the sensitivity of the approximations to parameter priors. For the experimental condition defined by $n = 64$, $r_{ct} = 8$, $N = 400$, and $\eta = 1.75$, we evaluated the approximations using three Dirichlet priors: $\alpha_{ijk} = 1$ (uniform in Θ_m); $\alpha_{ijk} = 1/r_i q_i$; and $\alpha_{ijk} = 0.1/r_i q_i$. The second and third priors are a special case of the priors described by Heckerman et al. (1995). The results are shown in Figure 8.

All approximations except the BIC/MDL ML were sensitive to the variation in priors.[7] This result demonstrates that it can be important to choose a prior carefully. In addition, it shows that the BIC/MDL approximation is inferior to the others in the sense that it is unresponsive to the prior.[8]

Recall one of the conditions used to derive the Laplace approximation: the MAP configuration $\tilde{\boldsymbol{\phi}}_m$ should lie away from the boundary of $\boldsymbol{\phi}_m$. We examined the sensitivity of the approximations to violations of this condition. In particular, we generated a model with $n = 64$ and $r_{ct} = 2$, assigning parameters according to the procedure in Section 4.3 with $\eta = 0$. That is, we generated a model with two identical multinomial mixtures. Then, with probability 0.1, we replaced each conditional probability $p(x_i^k|c^j, \boldsymbol{\phi}_m, \mathbf{m})$ with zero (a boundary value), and then renormalized each conditional distribution. If both probabilities $p(x_i^1|c^j, \boldsymbol{\phi}_m, \mathbf{m})$ and $p(x_i^2|c^j, \boldsymbol{\phi}_m, \mathbf{m})$ were set to zero, then we chose one of the probabilities at random and set it to one. We then generated a data set with $N = 400$. For comparison, we ran the experiment defined by $n = 64$, $r_{ct} = 2$, and $N = 400$, with parameters generated using $\eta = 1.75$. For both experimental conditions, the parameters for the hidden node were equal to 0.5. The interesting result, shown in Figure 9, is

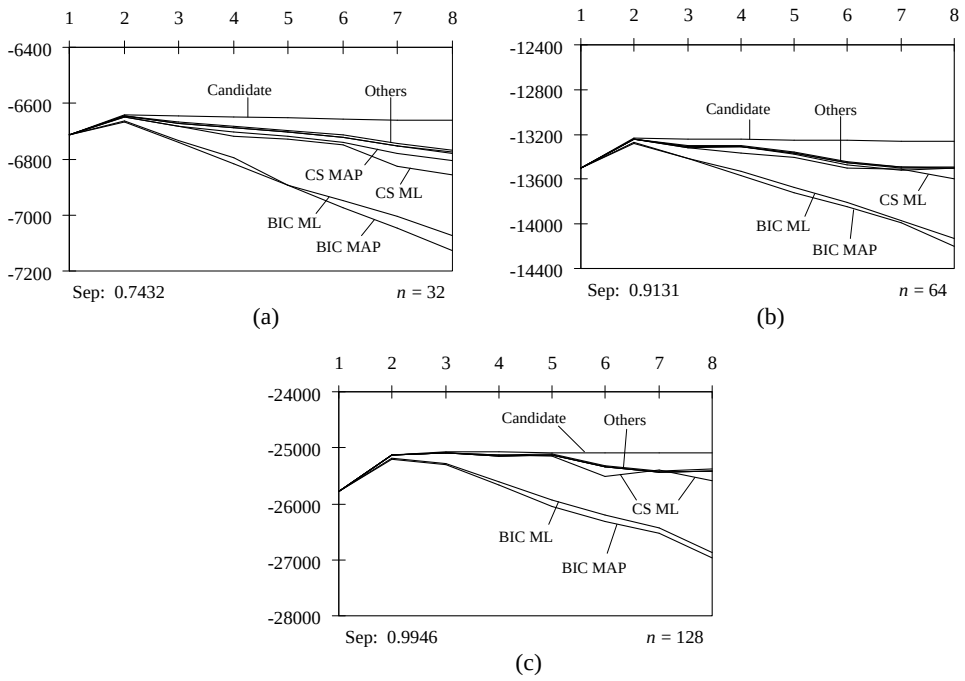

Figure 5. Sensitivity to n, the number of observed variables. Approximate log marginal likelihood versus r_c for models with $r_{ct} = 4$ and $\eta = 1.75$ and data sets with $N = 400$, when (a) $n = 32$, (b) $n = 64$, and (c) $n = 128$.

that the Cheeseman–Stutz MAP approximation yielded almost the same values as did the Monte-Carlo standard, whereas the other approximations performed as usual. That is, the Cheeseman–Stutz MAP approximation was more robust to violations of this assumption than were the other approximations, including the Laplace approximation. The result was reproducible for a variety of models.

This observation offers an explanation for the small N results in Figure 4, in which the Cheeseman–Stutz MAP approximation is more accurate than the other large-sample approximations. In particular, for small N, many observations for $\mathbf{X}$ that are possible are not realized in the data set. Consequently, many of the parameters in the MAP configuration for ϕ_m will be close to the boundary.

4.6. Computation times

As we have discussed, the accuracy of the approximations should be balanced against their computational costs. Figure 10 shows the costs for the experiment defined by $n = 64$, $r_{ct} = 8$, $N = 400$, and $\eta = 1.75$. The costs shown for the Candidate, Laplace, Block,

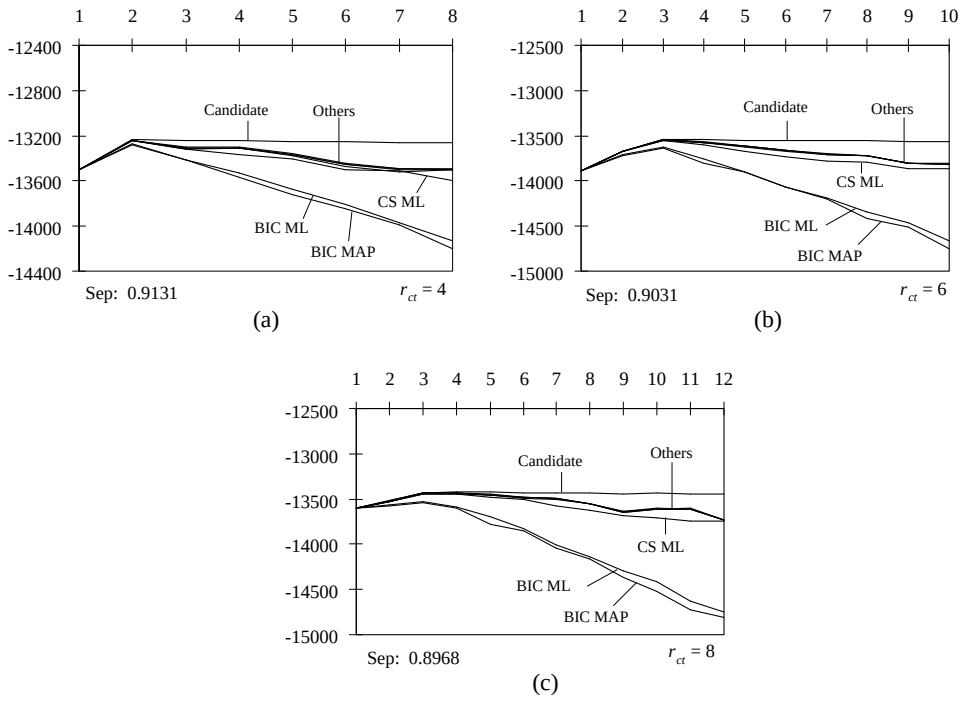

Figure 6. Sensitivity to r_{ct}, the number of classes in the true model. Approximate log marginal likelihood versus r_c for models with $n = 64$ and $\eta = 1.75$ and data sets with $N = 400$, when (a) $r_{ct} = 4$, (b) $r_{ct} = 6$, and (c) $r_{ct} = 8$.

Diagnonal, Cheeseman–Stutz, and BIC/MDL approximations exclude the computation of the MAP or ML configuration using the EM algorithm. This plot is in agreement with the computational complexities of the algorithms given in Section 3.6. Note that the EM algorithm dominates the cost of the Block, Diagonal, Cheeseman–Stutz, and BIC/MDL approximations. In contrast, the evaluation of the Hessian is more expensive than the cost of finding the MAP configuration using EM. Also, note that the Monte-Carlo standard is almost as efficient as the Laplace approximation for larger models.

4.7. Real-world data sets

To augment our experiments with synthetic data, we evaluated the various approximations on real-world data sets. We checked several data repositories, but could not locate data sets that involved discrete-variable clustering. Instead, we obtained classification data sets from the UCI Machine Learning Repository (Merz & Murphy, 1996) and discarded the known class information. We used the small soybean (Michalski & Chilausky, 1980), standard

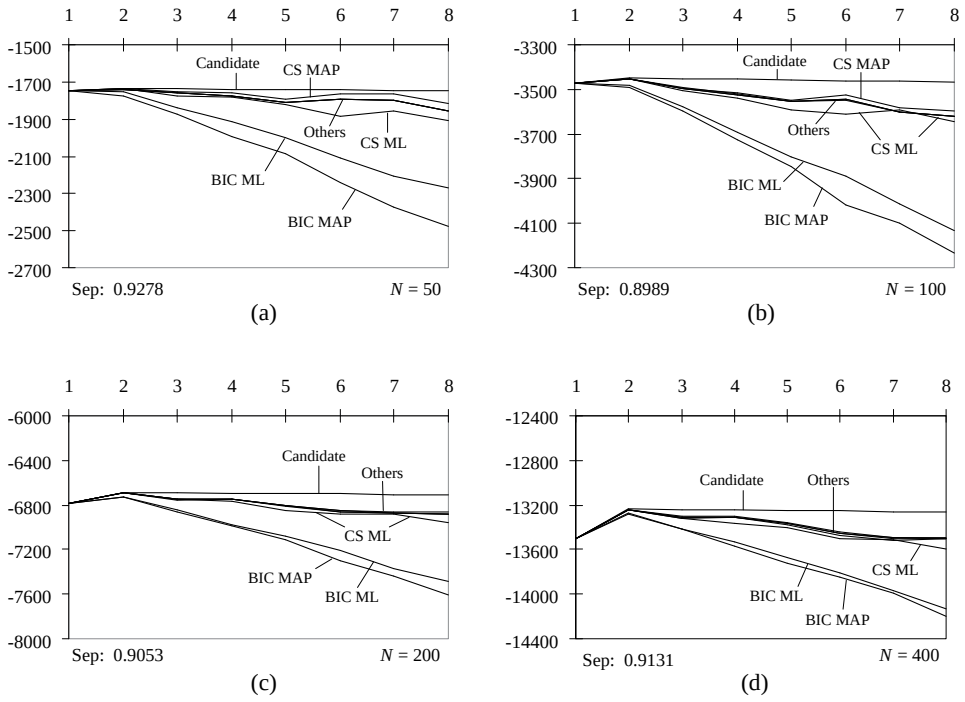

Figure 7. Sensitivity to N, sample size. Approximate log marginal likelihood versus r_c for models with $n = 64$, $r_{ct} = 4$, and $\eta = 1.75$, when (a) $N = 50$, (b) $N = 100$, (c) $N = 200$, and (d) $N = 400$.

audiology (Bareiss & Porter, 1987), and lung cancer (Hong & Yang, 1994) databases. For the audiology data set, where both training and test data were available, we merged these data sources.

The results, shown in Figure 11 and Table 3, are similar to those for synthetic data. In particular, the BIC/MDL approximation tended to peak early and fell off more sharply than did the other approximations. The large-sample approximations (except Cheeseman–Stutz MAP) fell off more rapidly than did the Candidate approximation for $r_c \geq r_{c*}$. The Cheeseman–Stutz approximation was more accurate when the MAP configuration was used, whereas the BIC/MDL approximation was more accurate when the ML configuration was used. For the audiology data set, most approximations selected only two classes, far less than the specified number. Nonetheless, there were only two classes in the data set with more than 20 instances.

There were three deviations from the studies with synthetic data that occurred in the evaluation of all three data sets. First, there were some values of r_c for which the Laplace and/or Block approximation could not be computed, because the determinant of the Hessian (or block) was negative. Second, the Cheeseman–Stutz MAP approximation was more

Table 1. Errors in model selection—mean (s.d.) over five data sets.

Conditions			Δr_{c*}						
n	r_{ct}	N	Laplace	Block	Diagonal	CS MAP	CS ML	BIC MAP	BIC ML
32	4	400	0	0	0	0	0	0	0
64	4	400	0	0	0	0	0	0	0
128	4	400	0.2(0.4)	0.2(0.4)	0.2(0.4)	0.2(0.4)	0.2(0.4)	1.0(0)	1.0(0)
64	4	400	0	0	0	0	0	0	0
64	6	400	0.4(0.5)	0.4(0.5)	0.4(0.5)	0.4(0.5)	0.4(0.5)	0.4(0.5)	0.4(0.5)
64	8	400	0.2(0.4)	0.2(0.4)	0.2(0.4)	0.6(0.5)	1.0(0.7)	1.0(0.7)	1.0(0.7)
64	4	50	0.2(0.4)	0.2(0.4)	0.2(0.4)	0	0.4(0.5)	0.6(0.5)	0.6(0.5)
64	4	100	0	0	0	0	0.2(0.4)	0.8(0.5)	0.6(0.5)
64	4	200	0	0	0	0	0	0	0
64	4	400	0	0	0	0	0	0	0

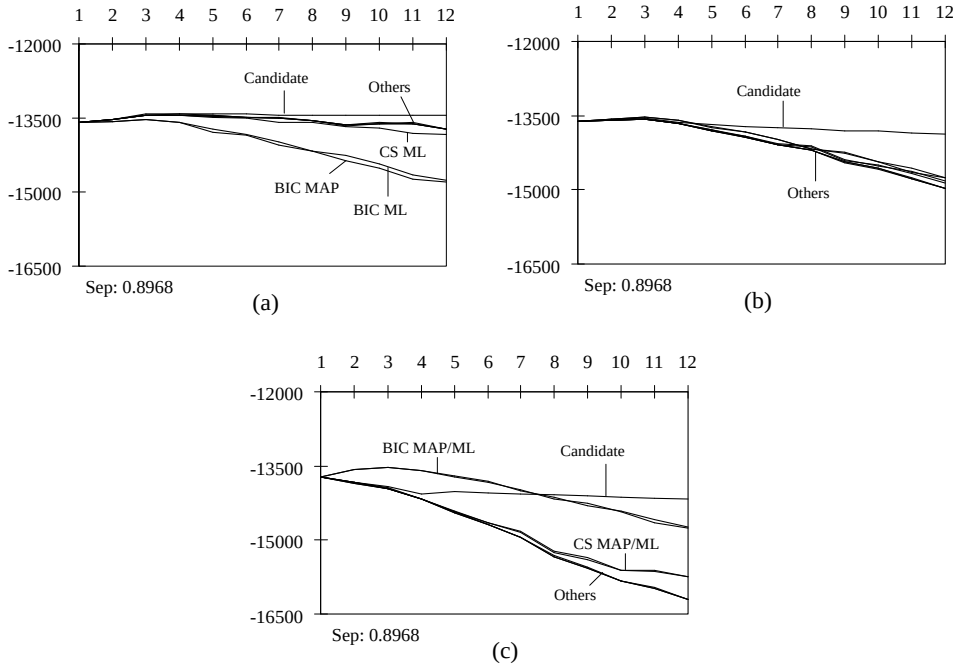

Figure 8. Sensitivity to parameter priors. In each experiment, $n = 64, r_{ct} = 8, N = 400$, and $\eta = 1.75$. The Dirichlet priors are given by (a) $\alpha_{ijk} = 1$, (b) $\alpha_{ijk} = 1/r_i q_i$, and (c) $\alpha_{ijk} = 0.1/r_i q_i$.

accurate than the other large-sample approximations. Third, many of the parameters were near the boundary in the MAP configurations for $\boldsymbol{\phi}_m$. This last observation explains the first two. In particular, when parameters are near the boundary, $\log p(\boldsymbol{\phi}_m|D, \mathbf{m})$ need not

Table 2. Errors in model averaging. Log Bayes factors given by the Candidate (C) and Laplace (L) methods are shown.

Conditions			$\log p(D\|\mathbf{m}_1)/p(D\|\mathbf{m}_2)$									
			data set 1		data set 2		data set 3		data set 4		data set 5	
n	r_{ct}	N	C	L	C	L	C	L	C	L	C	L
32	4	400	0.5	18.5	1.7	17.7	3.5	23.5	4.8	17.7	4.2	19.0
64	4	400	4.9	65.8	4.0	32.4	3.5	15.1	7.7	16.3	4.1	24.7
128	4	400	12.6	113	3.0	59.2	8.0	280	3.6	77.5	0.1	- 6.3
64	4	400	4.9	65.8	4.0	32.4	3.5	15.1	7.7	16.3	4.1	24.7
64	6	400	1.8	0.0	6.0	28.7	0	9.3	3.1	0.0	0.6	4.8
64	8	400	0.3	16.9	1.0	14.8	5.7	20.3	1.6	-4.6	10.3	74.4
64	4	50	3.2	14.5	1.8	-2.1	2.2	20.1	19.3	24.0	2.2	41.2
64	4	100	2.8	39.1	3.2	42.2	2.8	29.8	3.0	41.9	3.1	23.2
64	4	200	2.9	55.4	2.2	35.9	4.3	22.5	3.5	31.8	3.4	32.8
64	4	400	4.9	65.8	4.0	32.4	3.5	15.1	7.7	16.3	4.1	24.7

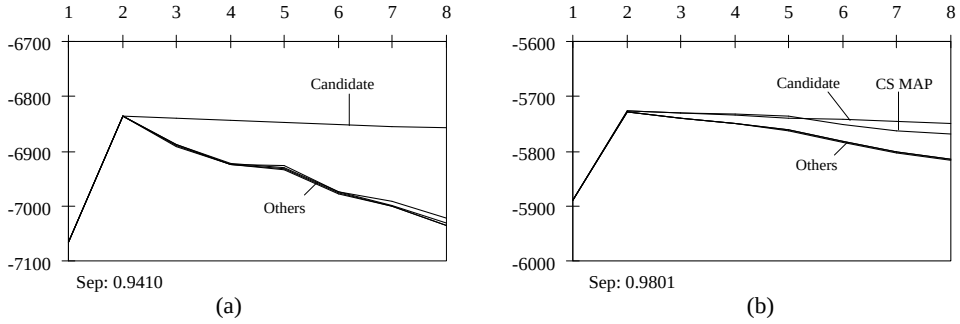

Figure 9. Sensitivity to parameters at the boundary. In both experiments, $n = 64$, $r_{ct} = 2$, $N = 400$. (a) Parameters are generated with $\eta = 1.75$. (b) 10% of the parameters $p(x_i^k|c^j, \phi_m, \mathbf{m})$ are set to zero.

be concave down around $\tilde{\phi}_m$. Furthermore, as we saw in Section 4.5, the Cheeseman–Stutz MAP is more robust to situations in which parameters in the MAP configuration are near the boundary.

5. Discussion

We have evaluated the accuracy and efficiency of the Laplace, Block-Diagnonal, Diagonal, Cheeseman–Stutz, and BIC/MDL approximations for the marginal likelihood of naive-Bayes models with a hidden root node. In this evaluation, we used the Monte-Carlo Candidate method as a gold standard. From our experiments, we draw a number of conclusions:

- None of approximations are accurate when used for model averaging.

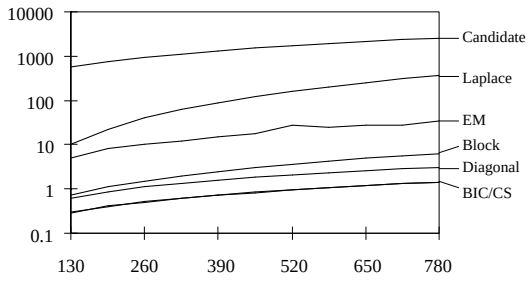

Figure 10. Computation time in seconds versus model dimension for the experimental condition $n = 64$, $r_{ct} = 8$, $N = 400$, and $\eta = 1.75$.

- All of the approximations, with the exception of BIC/MDL, are accurate for model selection.
- Among the accurate approximations, the Cheeseman–Stutz and Diagonal approximations are the most efficient.
- All of the approximations, with the exception of BIC/MDL, can be sensitive to the prior distribution over model parameters.
- The Cheeseman–Stutz approximation is more accurate when evaluated using the maximum a posteriori (MAP) configuration of the parameters, whereas the BIC/MDL approximation is more accurate when evaluated using the maximum likelihood (ML) configuration.
- The Cheeseman–Stutz approximation can be more accurate than the other approximations, including the Laplace approximation, in situations where the parameters in the MAP configuration are near a boundary.

Our findings are valid only for naive-Bayes models with a hidden root node, but these results are important, because they apply directly to probability-based clustering. Also, it seems likely that our results will extend to models for discrete variables where each variable that is unobserved has an observed Markov blanket. Under these conditions, each Bayesian inference required by the scoring functions (e.g., Equation 17) reduces to a naive-Bayes computation. Nonetheless, more extensive experiments are warranted to address models with more general structure and non-discrete distributions.

Although we have examined the computation of marginal likelihood for model averaging and model selection, we have not concentrated on how to handle the parameters once a model or set of models have been selected. If computation time is not an issue and one is concerned primarily with prediction, then a Monte-Carlo average over parameters is probably best (Neal, 1991). Nonetheless, one sometimes needs a fast model for prediction

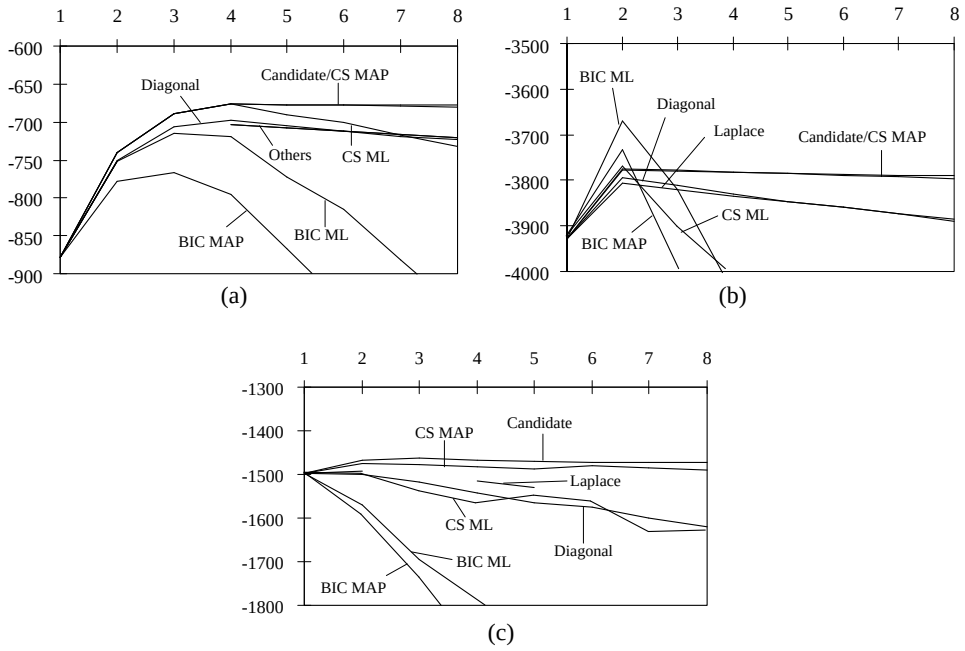

Figure 11. Plots of approximate log marginal likelihoods versus r_c for the (a) small soybean (b) standard audiology, and (c) lung cancer data sets.

Table 3. Number of classes selected by the approximations.

				r_{c*}					
data set	n	r_{ct}	N	Candidate	Laplace	Block	Diagonal	CS MAP	BIC ML
Small soybean	35	4	47	4	3–4	3–4	4	4	3
Audiology	70	24	226	2	2	?	2	2	2
Lung cancer	56	3	32	3	?	?	1	2	1

or one may want point values for the parameters to facilitate an understanding of the domain. What is best in these circumstances remains an open question.

Acknowledgments

We thank Wray Buntine, Dan Geiger, Michael Jordan, Daphne Koller, Yan LeCun, Chris Meek, Radford Neal, Adrian Raftery, Padhraic Smyth, Bo Thiesson, and Larry Wasserman for useful discussions. We also thank David MacKay for his suggestions regarding the Candidate method.

Notes

1. Throughout this paper, we use "efficiency" to refer to computational efficiency as opposed to statistical efficiency.
2. An equivalent criterion that is often used is $\log(p(\mathbf{m}|D)/p(\mathbf{m}_0|D)) = \log(p(\mathbf{m})/p(\mathbf{m}_0)) + \log(p(D|\mathbf{m})/p(D|\mathbf{m}_0))$. The ratio $p(D|\mathbf{m})/p(D|\mathbf{m}_0)$ is known as a *Bayes factor*.
3. One of the technical assumptions used to derive this approximation is that the prior distribution is non-zero around $\hat{\phi}_m$.
4. For this observation to hold, the Gibbs sampler must be *irreducible*. That is, the probability distribution $p(\mathbf{x})$ must be such that we can eventually sample any possible configuration of $\mathbf{X}$ given any possible initial configuration of $\mathbf{X}$. For example, if $p(\mathbf{x})$ contains no zero probabilities, then the Gibbs sampler will be irreducible.
5. This procedure was suggested by David MacKay (1996) in a personal communication.
6. Using Jensen et al.'s (1990) inference algorithm, only one inference is needed per expectation step.
7. Marginal likelihoods for $r_c = 1$ were sensitive to priors because we computed these values exactly using Equation 11.
8. In previous experiments (Chickering & Heckerman, 1996), we considered another large-sample approximation for the marginal likelihood suggested by Draper (1995). His approximation suffers from the same lack of sensitivity to the prior as does the BIC/MDL ML approximation.

References

Azevedo-Filho, A. & Shachter, R. (1994). Laplace's method approximations for probabilistic inference in belief networks with continuous variables. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 28–36). San Mateo, CA: Morgan Kaufmann.

Bareiss, E. & Porter, B. (1987). Protos: An exemplar-based learning apprentice. In *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 12–23). San Mateo, CA: Morgan Kaufmann.

Becker, S. & LeCun, Y. (1989). Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 Connectionist Models Summer School* (pp. 29–37). San Mateo, CA: Morgan Kaufmann.

Berger, J. (1985). *Statistical decision theory and Bayesian analysis*. Berlin: Springer.

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

Buntine, W. (1994a). Computing second derivatives in feed-forward networks: A review. *IEEE Transactions on Neural Networks, 5*, 480–488.

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

Buntine, W. (1996). A guide to the literature on learning graphical models. *IEEE Transactions on Knowledge and Data Engineering, 8*, 195–210.

Cheeseman, P. & Stutz, J. (1995). Bayesian classification (AutoClass): Theory and results. In Fayyad, U., Piatesky-Shapiro, G., Smyth, P., & Uthurusamy, R.(Eds.), *Advances in knowledge discovery and data mining*, pp. 153–180. Menlo Park, CA: AAAI Press.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association, 90*, 1313–1321.

Chickering, D. & 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* (pp. 158–168). San Mateo, CA: Morgan Kaufmann.

Clogg, C. (1995). Latent class models. In Arminger, G., Clogg, C., & Sobel, M. (Eds.), *Handbook of statistical modeling for the social and behavioral sciences*. Plenum Press, New York.

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

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

Draper, D. (1995). Assessment and propagation of model uncertainty (with discussion). *Journal of the Royal Statistical Society B, 57*, 45–97.

Geiger, D. & Heckerman, D. (1994). Learning Gaussian networks. In *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 235–243). San Mateo, CA: Morgan Kaufmann.

Geiger, D., Heckerman, D., & Meek, C. (1996). Asymptotic model selection for directed networks with hidden variables. In *Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence* (pp. 283–290). San Mateo, CA: Morgan Kaufmann.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6*, 721–742.

Gilks, W., Richardson, S., & Spiegelhalter, D. (1996). *Markov chain Monte Carlo in practice*. New York: Chapman and Hall.

Good, I. (1965). *The estimation of probabilities*. Cambridge, MA: MIT Press.

Gull, S. & Skilling, J. (1991). Quantified maximum entropy. MemSys5 user's manual. Tech. rep., M.E.D.C., 33 North End, Royston, SG8 6NR, England.

Haughton, D. (1988). On the choice of a model to fit data from an exponential family. *Annals of Statistics, 16*, 342–355.

Heckerman, D. (1995). A tutorial on learning Bayesian networks. Tech. rep. MSR-TR-95-06, Microsoft Research, Redmond, WA. Revised January, 1996.

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

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

Hong, Z. & Yang, J. (1994). Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition, 24*, 317–324.

Jeffreys, H. (1939). *Theory of probability*. Oxford University Press.

Jensen, F., Lauritzen, S., & Olesen, K. (1990). Bayesian updating in recursive graphical models by local computations. *Computational Statisticals Quarterly, 4*, 269–282.

Kass, R. & Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association, 90*, 773–795.

Kass, R., Tierney, L., & Kadane, J. (1988). Asymptotics in Bayesian computation. In Bernardo, J., DeGroot, M., Lindley, D., & Smith, A.(Eds.), *Bayesian statistics 3* (pp. 261–278). Oxford University Press.

Kass, R. & Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association, 90*, 928–934.

MacKay, D. (1992a). Bayesian interpolation. *Neural Computation, 4*, 415–447.

MacKay, D. (1992b). A practical Bayesian framework for backpropagation networks. *Neural Computation, 4*, 448–472.

MacKay, D. (1996). Choice of basis for the Laplace approximation. Tech. rep., Cavendish Laboratory, Cambridge, UK.

Madigan, D. & York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review, 63*, 215–232.

Meng, X. & Rubin, D. (1991). Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm. *Journal of the American Statistical Association, 86*, 899–909.

Merz, C. & Murphy, P. (1996). UCI repository of machine learning databases, www.ics.uci.edu/ ∼mlearn/mlrepository.html. Tech. rep., University of California, Irvine.

Michalski, R. & Chilausky, R. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems, 4*.

Neal, R. (1991). Bayesian mixture modeling by Monte Carlo simulation. Tech. rep. CRG-TR-91-2, Department of Computer Science, University of Toronto.

Neal, R. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Tech. rep. CRG-TR-93-1, Department of Computer Science, University of Toronto.

Raftery, A. (1994). Approximate Bayes factors and accounting for model uncertainty in generalized linear models. Tech. rep. 255, Department of Statistics, University of Washington.

Raftery, A. (1995). Bayesian model selection in social research. In Marsden, P. (Ed.), *Sociological methodology*. Cambridge, MA: Blackwells.

Raftery, A. (1996). *Hypothesis testing and model selection*, chap. 10. Chapman and Hall.
Rissanen, J. (1987). Stochastic complexity (with discussion). *Journal of the Royal Statistical Society, Series B*, *49*, 223–239 and 253–265.
Rubin, D. (1976). Inference and missing data. *Biometrika*, *3*, 581–592.
Russell, S., Binder, J., Koller, D., & Kanazawa, K. (1995). Local learning in probabilistic networks with hidden variables. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1146–1152). Morgan Kaufmann, San Mateo, CA.
Saul, L., Jaakkola, T., & Jordan, M. (1996). Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, *4*, 61–76.
Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.
Spiegelhalter, D., Dawid, A., Lauritzen, S., & Cowell, R. (1993). Bayesian analysis in expert systems. *Statistical Science*, *8*, 219–282.
Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, Prediction, and Search*. New York: Springer-Verlag.
Thiesson, B. (1997). Score and information for recursive exponential models with incomplete data. In *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann.

Received July 2, 1996
Accepted January 14, 1997
Final Manuscript July 30, 1997