1 Graphical models and variational methods

Zoubin Ghahramani and Matthew J. Beal
Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR
United Kingdom

The MIT Press
Cambridge, Massachusetts
London, England

1 Abstract

We review the use of variational methods of approximating inference and learning in probabilistic graphical models. In particular, we focus on variational approximations to the integrals required for Bayesian learning. For models in the conjugate-exponential family, a generalisation of the EM algorithm is derived that iterates between optimising hyperparameters of the distribution over parameters, and inferring the hidden variable distributions. These approximations make use of available propagation algorithms for probabilistic graphical models. We give two case studies of how the variational Bayesian approach can be used to learn model structure: inferring the number of clusters and dimensionalities in a mixture of factor analysers, and inferring the dimension of the state space of a linear dynamical system. Finally, importance sampling corrections to the variational approximations are discussed, along with their limitations.

2 Introduction

To design learning machines that reason about and act on the real world we need to represent uncertainty. Probability theory provides a language for representing uncertain beliefs and a calculus for manipulating these beliefs in a consistent manner [4, 28, 16]. However, the real world problems a machine may be faced with might involve hundreds or thousands of variables, and at first it may seem daunting to represent and manipulate joint distributions over all these variables. Fortunately, we can assume that of all possible direct dependencies between variables only a fraction are needed in most interesting problem domains. The dependencies and independencies between variables can be represented graphically, in the form of *probabilistic graphical models*. Such graphical models are not only a tool for visualising the relationships between variables but, by exploiting the conditional independence relationships, also provide a backbone upon which it has been possible to derive efficient message-propagating algorithms for updating the uncertain beliefs of the machine [28, 21, 18, 12]. This chapter focuses on learning and belief updating in models for which these are intractable despite the use of these efficient propagation algorithms. For such models one has to resort to approximate methods; we present approximations based on *variational methods*, which are closely related to *mean-field methods* in statistical physics.

Variational methods have been developed both for maximum likelihood (ML) learning and Bayesian learning. In section 3 we describe their use in ML learning, which is reviewed in more detail in [20]. Readers familiar with the lower-bound derivation of EM and the use of variational methods in ML learning can skip this section. In section 4, we motivate how the Bayesian approach of integrating over model parameters avoids overfitting and can be used to select model structures. Variational methods are used to approximate these intractable integrals. Section 5 considers models which fall in the conjugate-exponential class and presents the variational Bayesian EM algorithm, which generalises the maximum likelihood EM algorithm. Section 6 describes how the variational Bayesian algorithm can make use of propagation algorithms for graphical models. In section 7, we provide several example applications of variational methods to Bayesian inference of model structure. Section 8 discusses combining sampling methods with variational methods to estimate the quality of the variational bounds. Finally, we conclude with section 9. We assume that the reader is familiar with the basics of inference in probabilistic graphical models. For relevant tutorials he or she is referred to: [18, 12, 19, 30].

3 Variational methods for maximum likelihood learning

Variational methods have been used for approximate maximum likelihood learning in probabilistic graphical models with hidden variables. To understand their role it is instructive to derive the EM algorithm for maximum likelihood learning.

Consider a graphical model with hidden variables $\mathbf{x}$, observable variables $\mathbf{y}$, and parameters $\boldsymbol{\theta}$. ML learning seeks to maximize the likelihood, or equivalently the log likelihood, of a data set $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ as a function of $\boldsymbol{\theta}$:

$$\mathcal{L}(\boldsymbol{\theta}) = \ln P(Y|\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln P(\mathbf{y}_i|\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln \int d\mathbf{x}\; P(\mathbf{y}_i, \mathbf{x}|\boldsymbol{\theta}) \qquad (1)$$

where we have assumed the data is independent and identically distributed (iid). The integral (or sum) over $\mathbf{x}$ is required to obtain the marginal probability of the data. Maximising (1) directly is often difficult because the log of the integral can potentially couple all of the parameters of the model. Furthermore, for models with many hidden variables, the integral (or sum) over $\mathbf{x}$ can be intractable. We can simplify the problem of maximising $\mathcal{L}$ with respect to $\boldsymbol{\theta}$ by making use of the following insight. Any distribution $Q_{\mathbf{x}}(\mathbf{x})$ over the hidden variables defines a *lower bound* on $\mathcal{L}$. In fact, for each data point $\mathbf{y}_i$ we use a distinct distribution $Q_{\mathbf{x}_i}(\mathbf{x}_i)$ over the hidden variables to get the lower bound:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) = \sum_i \ln \int d\mathbf{x}_i P(\mathbf{y}_i, \mathbf{x}_i|\boldsymbol{\theta}) &= \sum_i \ln \int d\mathbf{x}_i \; Q_{\mathbf{x}_i}(\mathbf{x}_i) \frac{P(\mathbf{y}_i, \mathbf{x}_i|\boldsymbol{\theta})}{Q_{\mathbf{x}_i}(\mathbf{x}_i)} && (2) \\ &\geq \sum_i \int d\mathbf{x} \; Q_{\mathbf{x}_i}(\mathbf{x}_i) \ln \frac{P(\mathbf{y}_i, \mathbf{x}_i|\boldsymbol{\theta})}{Q_{\mathbf{x}_i}(\mathbf{x}_i)} && (3) \\ &= \mathcal{F}(Q_{\mathbf{x}_1}(\mathbf{x}_1), \ldots, Q_{\mathbf{x}_n}(\mathbf{x}_n), \boldsymbol{\theta}) && (4) \end{aligned}$$

where the inequality is known as Jensen's inequality and follows from the fact that the ln function is concave. Defining the *energy* of a global configuration $(\mathbf{x}, \mathbf{y})$ to be $-\ln P(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$, the lower bound $\mathcal{F} \leq \mathcal{L}(\boldsymbol{\theta})$ is the negative of a quantity known in statistical physics as the *free energy*: the expected energy under Q minus the entropy of Q [26], where we use Q to mean the set of all $Q_{\mathbf{x}_i}$. The Expectation-Maximization (EM) algorithm [2, 5] alternates between maximising $\mathcal{F}$ with respect to the $Q_{\mathbf{x}_i}$ and $\boldsymbol{\theta}$, respectively, holding the other fixed. Starting from some initial parameters $\boldsymbol{\theta}^0$:

$$\textbf{E step:} \qquad Q_{\mathbf{x}_i}^{k+1} \leftarrow \arg\max_{Q_{\mathbf{x}_i}} \; \mathcal{F}(Q, \boldsymbol{\theta}^k), \qquad \forall\, i \tag{5}$$

$$\textbf{M step:} \qquad \boldsymbol{\theta}^{k+1} \leftarrow \arg\max_{\boldsymbol{\theta}} \; \mathcal{F}(Q^{k+1}, \boldsymbol{\theta}) \tag{6}$$

It is easy to see that the maximum in the E step is obtained by setting $Q_{\mathbf{x}_i}^{k+1}(\mathbf{x}) = P(\mathbf{x}|\mathbf{y}_i, \boldsymbol{\theta}^k)$, at which point the bound becomes an equality: $\mathcal{F}(Q^{k+1}, \boldsymbol{\theta}^k) = \mathcal{L}(\boldsymbol{\theta}^k)$. The maximum in the M step is obtained by minimising the expected energy term in (3), since the entropy of Q does not depend on $\boldsymbol{\theta}$:

$$\textbf{M step:} \quad \boldsymbol{\theta}^{k+1} \leftarrow \arg\max_{\boldsymbol{\theta}} \sum_i \int d\mathbf{x} \; P(\mathbf{x}|\mathbf{y}_i, \boldsymbol{\theta}^k) \ln P(\mathbf{x}, \mathbf{y}_i|\boldsymbol{\theta}).$$

Since $\mathcal{F} = \mathcal{L}$ at the beginning of each M step, and since the E step does not change $\boldsymbol{\theta}$, we are guaranteed not to decrease the likelihood after each combined EM step.

It is usually not necessary to evaluate the posterior distribution $P(\mathbf{x}|\mathbf{y}_i, \boldsymbol{\theta}^k)$ explicitly. For example, if $\ln P(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ contains both hidden and observed variables in a Bayesian network, it can be factored as the sum of log probabilities of each node given its parents.[1] Therefore, the quantities required for the M step are the expected values, under the posterior distributions $P(\mathbf{x}|\mathbf{y}_i, \boldsymbol{\theta}^k)$, of the sufficient statistics required for ML estimation in the complete data case.

For many models, especially those with multiple hidden variables forming a *distributed representation* of the observed variables, even these sufficient statistics are intractable to compute [24, 37, 13, 11, 10]. In the E step, rather than optimising $\mathcal{F}$ over all Q, we constrain Q to be of a particular form, for example factorised. We can still optimise $\mathcal{F}$ as a functional of constrained distributions Q using calculus of variations. This is the key step of variational approximations, and we return to it soon. Once this optimisation has been performed, we use the expected sufficient statistics with respect to Q, which can presumably be computed tractably, in the M step.

Maximising $\mathcal{F}$ with respect to $Q_{\mathbf{x}_i}$ is equivalent to minimising the following quantity:

$$\int d\mathbf{x} \; Q_{\mathbf{x}_i}(\mathbf{x}_i) \; \ln \frac{Q_{\mathbf{x}_i}(\mathbf{x}_i)}{P(\mathbf{x}|\mathbf{y}_i, \boldsymbol{\theta})}$$

which is the Kullback-Leibler (KL) divergence measuring the (asymmetric) difference between $Q_{\mathbf{x}_i}$ and the true posterior. Choosing $Q_{\mathbf{x}_i}$ to have easily computed moments, and if $\ln P$ is a polynomial in $\mathbf{x}$, we can compute the KL-divergence up to a constant and more importantly we can take its derivatives to minimise it with respect to the parameters of $Q_{\mathbf{x}_i}$.

The E step of this *variational EM* therefore consists of a sub-loop in

1 One of the defining properties of Bayesian networks is that the joint probability of all variables $P(z_1, \ldots, z_n)$ can be factored as $\prod_{i=1}^{n} P(z_i|z_{pa(i)})$ where $z_{pa(i)}$ is the set of variables whose nodes are parents of i in the network.

which the $Q_{\mathbf{x}_i}$ is optimised. We can often do this by taking derivatives with respect to the parameters of $Q_{\mathbf{x}_i}$ and iteratively solving the fixed point equations. For approximations where $Q_{\mathbf{x}_i}$ is fully factorised, i.e. $Q_{\mathbf{x}_i}(\mathbf{x}_i) = \prod_{j=1}^{m} Q_{x_{ij}}(x_{ij})$, these fixed point equations are called *mean-field equations* by analogy to such methods in statistical physics. Examples of these variational approximations can be found in [31, 6, 15, 11].

4 Variational methods for Bayesian learning

Maximum likelihood methods suffer from the problem that that they fail to take into account model complexity, which is, from an information theoretic view, the cost of coding the model parameters. Not penalising more complex models leads to overfitting and the inability to determine the best model size and structure. While it is possible to use cross-validation for simple searches over model size and structures—for example, if the search is limited to a single parameter that controls the model 'complexity'—for more general searches cross-validation is computationally prohibitive. Bayesian approaches overcome overfitting and learn model structure by treating the parameters $\boldsymbol{\theta}$ as unknown random variables and averaging over the ensemble of models one would obtain by sampling from $\boldsymbol{\theta}$:

$$P(Y|\mathcal{M}) = \int d\boldsymbol{\theta}\, P(Y|\boldsymbol{\theta}, \mathcal{M}) P(\boldsymbol{\theta}|\mathcal{M}). \tag{7}$$

$P(Y|\mathcal{M})$ is the *evidence* or *marginal likelihood* for a data set Y assuming model $\mathcal{M}$, and $P(\boldsymbol{\theta}|\mathcal{M})$ is the prior distribution over parameters. Integrating out parameters penalises models with more degrees of freedom since these models can *a priori* model a larger range of data sets. This property of Bayesian integration has been called Ockham's razor, since it favors simpler explanations (models) for the data over complex ones [17, 22]. The overfitting problem is avoided simply because no parameter in the pure Bayesian approach is actually *fit* to the data. Having more parameters imparts an advantage in terms of the ability to model the data, but this is offset by the cost of having to code that parameter under the prior [14].

Along with the prior over parameters, a Bayesian approach to learning starts with some prior knowledge or assumptions about the model structure—the set of arcs in the Bayesian network. This initial knowledge is represented in the form of a prior probability distribution over model structures, and is updated using the data to obtain a posterior distribution over models and parameters. More formally, assuming a prior distribution over models structures $P(\mathcal{M})$ and a prior distribution over parameters for each model structure $P(\boldsymbol{\theta}|\mathcal{M})$, observing the data set Y induces a posterior distribution over models given by Bayes rule:

$$P(\mathcal{M}|Y) = \frac{P(\mathcal{M})P(Y|\mathcal{M})}{P(Y)} \tag{8}$$

The most probable model or model structure is the one that maximises $P(\mathcal{M}|Y)$.

For a given model structure, we can also compute the posterior distribution over the parameters:

$$P(\boldsymbol{\theta}|Y, \mathcal{M}) = \frac{P(Y|\boldsymbol{\theta}, \mathcal{M})P(\boldsymbol{\theta}|\mathcal{M})}{P(Y|\mathcal{M})}.$$

which allows us to quantify our uncertainty about parameter values after observing the data. The density at a new data point $\mathbf{y}$ is obtained by averaging over both the uncertainty in the model structure and in the parameters,

$$P(\mathbf{y}|Y) = \int d\boldsymbol{\theta}\, d\mathcal{M}\ P(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M}, Y)P(\boldsymbol{\theta}|\mathcal{M}, Y)P(\mathcal{M}|Y)$$

This is known as the *predictive distribution*.

While Bayesian theory in principle avoids the problems of overfitting and can be used to do model selection and averaging, in practice it is often computationally and analytically intractable to perform the required integrals. *Markov chain Monte Carlo* (MCMC) methods can be used to approximate these integrals by sampling [25]. The main criticism of MCMC methods is that they are slow and it is usually difficult to assess convergence. Furthermore, the posterior density over parameters, $P(\boldsymbol{\theta}|Y, \mathcal{M})$ which captures all information inferred from the data about the parameters, is stored as a set of samples, which can be inefficient.

Another approach to Bayesian integration is the *Laplace approximation* which makes a local Gaussian approximation around a maximum *a posteriori* (MAP) parameter estimate [22]. These approximations are based on large data limits and can be poor, particularly for small data sets (for which, in principle, the advantages of Bayesian integration over ML are largest). Local Gaussian approximations are also poorly suited to bounded or positive parameters such as the mixing proportions of the mixture model. Finally, the Gaussian approximation requires computing or approximating the Hessian at the MAP estimate, which can be computationally costly.

Variational methods can be used for approximate the integrals required for Bayesian learning. The basic idea is to simultaneously approximate the distribution over both hidden states and parameters with a simpler distribution, usually by assuming the hidden states and parameters are independent. More specifically, in exactly the same way as the log likelihood is lower bounded in the derivation of EM (3), the log evidence can be lower bounded by applying Jensen's inequality:

$$
\begin{aligned}
&\ln P(Y|\mathcal{M}) \\
&\quad = \ln \int d\boldsymbol{\theta}\ P(Y, \boldsymbol{\theta}|\mathcal{M}) && (9)\\
&\quad \geq \iint d\boldsymbol{\theta}\ dX\ Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) Q_X(X) \ln \frac{P(Y, X, \boldsymbol{\theta}|\mathcal{M})}{Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) Q_X(X)} && (10)\\
&\quad = \int d\boldsymbol{\theta}\ Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \left[\int dX Q_X(X) \ln \frac{P(Y, X|\boldsymbol{\theta}, \mathcal{M})}{Q_X(X)} + \ln \frac{P(\boldsymbol{\theta}|\mathcal{M})}{Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \right] && (11)\\
&\quad = \mathcal{F}(Q_X(X), Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) && (12)\\
&\quad = \mathcal{F}(Q_{\mathbf{x}_1}(\mathbf{x}_1), \ldots, Q_{\mathbf{x}_n}(\mathbf{x}_n), Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})) && (13)
\end{aligned}
$$

The last equality follows from the fact that the observed data is iid. The variational Bayesian approach iteratively maximises $\mathcal{F}$ as a functional of the free distributions, $Q_X(X)$ and $Q(\boldsymbol{\theta})$. From (11) we can see that this maximisation is equivalent to minimising the KL divergence between $Q_X(X)\ Q(\boldsymbol{\theta})$ and the joint posterior over hidden states and parameters $P(X, \boldsymbol{\theta}|Y, \mathcal{M})$. Note the similarity between (4) and (13). While we maximise the former with respect to hidden variable distributions and the parameters, the latter we maximise w.r.t. hidden variable distributions and a parameter *distribution*.

This approach was first proposed for one-hidden layer neural networks (which have no hidden state) by Hinton and van Camp (1993) using the restriction that $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is Gaussian. The term *ensemble learning* was used to describe the method since it fits an ensemble of models, each with its own parameters. It has since been applied to various other models with hidden states and no restrictions on $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $Q_{\mathbf{x}_i}(\mathbf{x}_i)$ other than the assumption that they factorise in some way [36, 23, 3, 1, 8].

With only these factorisation assumptions, free-form optimisation with respect to the distributions $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $Q_{\mathbf{x}_i}(\mathbf{x}_i)$ is done using calculus of variations, and often results in a modified EM-like algorithm.

5 Conjugate-Exponential Models

We consider variational Bayesian learning in models that satisfy two conditions:

Condition (1). *The complete data likelihood is in the exponential family:*

$$P(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = f(\mathbf{x}, \mathbf{y})\; g(\boldsymbol{\theta}) \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}, \mathbf{y})\right\}$$

where $\boldsymbol{\phi}(\boldsymbol{\theta})$ is the vector of natural parameters, and $\mathbf{u}$ and f and g are the functions that define the exponential family.

The list of latent-variable models of practical interest with complete-data likelihoods in the exponential family is very long. We mention a few: Gaussian mixtures, factor analysis, hidden Markov models and extensions, switching state-space models, Boltzmann machines, and discrete-variable belief networks.[2] Of course, there are also many as yet undreamed-of models combining Gaussian, Gamma, Poisson, Dirichlet, Wishart, Multinomial, and other distributions.

Condition (2). *The parameter prior is conjugate to the complete data likelihood:*

$$P(\boldsymbol{\theta}|\eta, \boldsymbol{\nu}) = h(\eta, \boldsymbol{\nu})\; g(\boldsymbol{\theta})^\eta \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \boldsymbol{\nu}\right\}$$

where η and $\boldsymbol{\nu}$ are hyperparameters of the prior.

Condition (2) in fact usually implies condition (1). In general the exponential families are the only classes of distributions that have natural conjugate prior distributions because they are the only distributions with a fixed number of sufficient statistics apart from some irregular cases. From the definition of conjugacy it is easy to see that the hyperparameters of a conjugate prior can be interpreted as the number (η) and values ($\boldsymbol{\nu}$) of pseudo-observations under the corresponding likelihood. We call models that satisfy conditions (1) and (2) *conjugate-exponential.*

---

2 Models whose complete-data likelihood is not in the exponential family (such as ICA with the logistic nonlinearity, or logistic regression) can often be approximated by models in the exponential family with additional hidden variables.

In Bayesian inference we want to determine the posterior over parameters and hidden variables $P(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}, \eta, \boldsymbol{\nu})$. In general this posterior is *neither* conjugate nor in the exponential family. This motivates the use of variational methods, which we described in the previous section. We provide several general results for variational Bayesian learning of conjugate-exponential models, with no proof. The proofs and additional detail will be provided in the journal version of this chapter (in preparation).

Theorem 1 *Given an iid data set $Y = \{\mathbf{y}_1, \ldots \mathbf{y}_n\}$, if the model satisfies conditions (1) and (2), then at the maxima of $\mathcal{F}(Q(X), Q(\boldsymbol{\theta}))$:*

(a) *$Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is conjugate and of the form:*

$$Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = h(\tilde{\eta}, \tilde{\boldsymbol{\nu}}) g(\boldsymbol{\theta})^{\tilde{\eta}} \exp\left\{\boldsymbol{\phi}(\boldsymbol{\theta})^\top \tilde{\boldsymbol{\nu}}\right\}$$

where

$$\tilde{\eta} = \eta + n$$

$$\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + \sum_{i=1}^{n} \overline{\mathbf{u}}(\mathbf{x}_i, \mathbf{y}_i),$$

and $\overline{\mathbf{u}}(\mathbf{x}_i, \mathbf{y}_i) = \langle \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i) \rangle_Q$, using $\langle \cdot \rangle_Q$ to denote expectation under Q.

(b) *$Q_X(X) = \prod_{i=1}^n Q_{\mathbf{x}_i}(\mathbf{x}_i)$ and $Q_{\mathbf{x}_i}(\mathbf{x}_i)$ is of the same form as the known parameter posterior:*

$$Q_{\mathbf{x}_i}(\mathbf{x}_i) \propto f(\mathbf{x}_i, \mathbf{y}_i) \exp\left\{\overline{\boldsymbol{\phi}}(\boldsymbol{\theta})^\top \mathbf{u}(\mathbf{x}_i, \mathbf{y}_i)\right\} = P(\mathbf{x}_i | \mathbf{y}_i, \overline{\boldsymbol{\phi}}(\boldsymbol{\theta}))$$

where $\overline{\boldsymbol{\phi}}(\boldsymbol{\theta}) = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_Q$.

Since $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $Q_{\mathbf{x}_i}(\mathbf{x}_i)$ are coupled, (a) and (b) do not provide an analytic solution to the minimisation problem. We therefore solve the optimisation problem numerically by iterating between the fixed point equations given by (a) and (b), and we obtain the following variational Bayesian generalisation of the EM algorithm:

VE Step: *Compute the expected sufficient statistics $\mathbf{t}(Y) = \sum_i \overline{\mathbf{u}}(\mathbf{x}_i, \mathbf{y}_i)$ under the hidden variable distributions $Q_{\mathbf{x}_i}(\mathbf{x}_i)$.*

VM Step: *Compute the expected natural parameters $\overline{\boldsymbol{\phi}}(\boldsymbol{\theta})$ under the parameter distribution given by $\tilde{\eta}$ and $\tilde{\boldsymbol{\nu}}$.*

This reduces to the EM algorithm if we restrict the parameter density to a point estimate (i.e. Dirac delta function), $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$, in which case the M step involves re-estimating $\boldsymbol{\theta}^*$.

Note that unless we make the assumption that the parameters and hidden variables factorise, we will not generally obtain the further hidden variable factorisation over n in (b). In that case, the distributions of $\mathbf{x}_i$ and $\mathbf{x}_j$ will be coupled for all cases i, j in the data set, greatly increasing the overall computational complexity of inference.

6 Belief Networks and Markov Networks

The above result can be used to derive variational Bayesian learning algorithms for exponential family distributions that fall into two important special classes. [3]

Corollary 1: Conjugate-Exponential Belief Networks. *Let $\mathcal{M}$ be a conjugate-exponential model with hidden and visible variables $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ that satisfy a belief network factorisation. That is, each variable z_j has parents $\mathbf{z}_{p_j}$ and $P(\mathbf{z}|\boldsymbol{\theta}) = \prod_j P(z_j|\mathbf{z}_{p_j}, \boldsymbol{\theta})$. Then the approximating joint distribution for $\mathcal{M}$ satisfies the same belief network factorisation:*

$$Q_{\mathbf{z}}(\mathbf{z}) = \prod_j Q(z_j|\mathbf{z}_{p_j}, \tilde{\boldsymbol{\theta}})$$

where the conditional distributions have exactly the same form as those in the original model but with natural parameters $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}(\boldsymbol{\theta})$. Furthermore, with the modified parameters $\tilde{\boldsymbol{\theta}}$, the expectations under the approximating posterior $Q_{\mathbf{x}}(\mathbf{x}) \propto Q_{\mathbf{z}}(\mathbf{z})$ required for the VE Step can be obtained by applying the **belief propagation** *algorithm if the network is singly connected and the* **junction tree** *algorithm if the network is multiply-connected.*

This result is somewhat surprising as it shows that it is possible to infer the hidden states tractably while integrating over an ensemble of model parameters. This result generalises the derivation of variational learning for HMMs in [23], which uses the forward-backward algorithm as a subroutine.

Theorem 2: Markov Networks. *Let $\mathcal{M}$ be a model with hidden and visible variables $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ that satisfy a Markov network factorisation. That is, the joint density can be written as a product of clique-potentials ψ_j, $P(\mathbf{z}|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) \prod_j \psi_j(C_j, \boldsymbol{\theta})$, where each clique C_j is a subset of the variables in $\mathbf{z}$. Then the approximating joint distribution for $\mathcal{M}$ satisfies the same Markov network factorisation:*

$$Q_{\mathbf{z}}(\mathbf{z}) = \tilde{g} \prod_j \overline{\psi}_j(C_j)$$

where $\overline{\psi}_j(C_j) = \exp\{\langle \ln \psi_j(C_j, \boldsymbol{\theta})\rangle_Q\}$ are new clique potentials obtained by averaging over $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, and $\tilde{g}$ is a normalisation constant. Further-

3 A tutorial on belief networks and Markov networks can be found in [28].

more, the expectations under the approximating posterior $Q_{\mathbf{x}}(\mathbf{x})$ required for the VE Step can be obtained by applying the junction tree algorithm.

Corollary 2: Conjugate-Exponential Markov Networks. *Let $\mathcal{M}$ be a conjugate-exponential Markov network over the variables in $\mathbf{z}$. Then the approximating joint distribution for $\mathcal{M}$ is given by $Q_{\mathbf{z}}(\mathbf{z}) = \tilde{g}\prod_j \psi_j(C_j, \tilde{\boldsymbol{\theta}})$, where the clique potentials have exactly the same form as those in the original model but with natural parameters $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}}(\boldsymbol{\theta})$.*

For conjugate-exponential models in which belief propagation and the junction tree algorithm over hidden variables is intractable further applications of Jensen's inequality can yield tractable factorisations in the usual way [20].

7 Examples

In this section we provide several examples of the variational Bayesian learning algorithm and show how the algorithm can be used to learn the structure of the model. We discuss two models in detail—mixtures of factor analysers and linear dynamical systems—and then briefly review several other models.

Mixtures of factor analysers

A factor analyser is a linear generative model that assumes the data was generated from zero-mean identity-covariance Gaussian distributed factors $\mathbf{x}$:

$$\mathbf{y} = \Lambda\mathbf{x} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon}$ is Gaussian noise with diagonal covariance matrix Ψ. Integrating out the factors $\mathbf{x}$ and noise, we get that $P(\mathbf{y}|\Lambda, \Psi)$ is zero mean Gaussian with covariance matrix $\Lambda\Lambda^\top + \Psi$. Generally, the vector of factors $\mathbf{x}$ is k-dimensional and $k < p$, where p is the dimensionality of the observation vectors $\mathbf{y}$, so factor analysis corresponds to fitting the covariance matrix of $\mathbf{y}$ vector with fewer than $p(p + 1)/2$ degrees of freedom.

A mixture of factor analysers (MFA) models the density for $\mathbf{y}$ as a weighted average of factor analyser densities

$$P(\mathbf{y}|\Lambda, \Psi, \boldsymbol{\pi}) = \sum_{s=1}^{S} P(s|\boldsymbol{\pi})P(\mathbf{y}|s, \Lambda^s, \Psi), \tag{14}$$

where $\boldsymbol{\pi}$ is the vector of mixing proportions, s is a discrete indicator variable, and Λ^s is the factor loading matrix for factor analyser s which includes a mean vector for $\mathbf{y}$.

By exploiting the factor analysis parameterisation of covariance matrices, a mixture of factor analysers can be used to fit a mixture of Gaussians to correlated high dimensional data without requiring $O(p^2)$ parameters or undesirable compromises such as axis-aligned covariance matrices. In an MFA each Gaussian cluster has intrinsic dimensionality k (or k_s if the dimensions are allowed to vary across clusters). The mixture of factor analysers therefore simultaneously tries to solve both a clustering problem and multiple local dimensionality reduction problems under Gaussian assumptions. When Ψ is a multiple of the identity the model becomes a mixture of probabilistic principal components analysis (PCA). Tractable maximum likelihood procedures for fitting MFA and MPCA models can be derived from the EM algorithm [9, 35].

Since $P(s|\boldsymbol{\pi})$ is multinomial, and both $P(\mathbf{x})$ and $P(\mathbf{y}|\mathbf{x}, s, \Lambda, \Psi)$ are Gaussian, the model satisfies condition (1), that is, it has a complete data likelihood in the exponential family. Note that if we were to integrate out $\mathbf{x}$ and sum over s the *marginal likelihood* of $P(\mathbf{y}|\Lambda, \Psi, \boldsymbol{\pi})$ is *not* in the exponential family; however, we need not worry about this.

Starting from (14), the evidence for the Bayesian MFA is obtained by averaging the likelihood under priors for the parameters (which have their own hyperparameters):

$$\begin{aligned} P(Y) \quad = \quad & \int d\boldsymbol{\pi} P(\boldsymbol{\pi}|\alpha) \int d\boldsymbol{\nu} P(\boldsymbol{\nu}|a, b) \int d\Lambda \; P(\Lambda|\boldsymbol{\nu}) \cdot \\ & \prod_{n=1}^{N} \left[\sum_{s^n=1}^{S} P(s^n|\boldsymbol{\pi}) \int d\mathbf{x}^n P(\mathbf{x}^n) P(\mathbf{y}^n|\mathbf{x}^n, s^n, \Lambda^s, \Psi) \right]. \qquad (15) \end{aligned}$$

Here $\{\alpha, a, b, \Psi\}$ are hyperparameters[4], and $\boldsymbol{\nu}$ are precision parameters (i.e. inverse variances) for the columns of Λ. We have dropped the conditioning on model class, $\mathcal{M}$, although this should be understood to be implicit in what follows. The conditional independence relations between the variables in this model are shown graphically in the usual belief network representation in Figure 1.

To satisfy condition (2) we choose conjugate priors. We choose $P(\boldsymbol{\pi}|\alpha)$ to be symmetric Dirichlet, which is conjugate to the multinomial

4 We currently do not integrate out Ψ, although this can also be done.

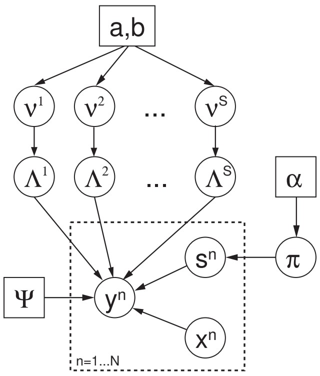

Figure 1
Generative model for variational Bayesian mixture of factor analysers. Circles denote random variables, solid rectangles denote hyperparameters, and the dashed rectangle shows the plate (i.e. repetitions) over the data.

$P(s|\boldsymbol{\pi})$. The prior for the factor loading matrix plays a key role in this model. Each component of the mixture has a Gaussian prior $P(\Lambda^s|\boldsymbol{\nu}^s)$, where each element of the vector $\boldsymbol{\nu}^s$ is the precision of a *column* of Λ. If one of these precisions $\boldsymbol{\nu}_l^s \to \infty$, then the outgoing weights for factor $\mathbf{x}_l$ will go to zero, which allows the model to reduce the intrinsic dimensionality of $\mathbf{x}$ if the data does not warrant this added dimension. A previous use of such Gaussian priors for intrinsic dimensionality reduction can be found in [3] for Bayesian PCA. These Gaussian priors are called *automatic relevance determination* (ARD) priors as they were used by MacKay and Neal to do relevant input variable selection in neural networks [27].

To avoid overfitting it is important to integrate out all parameters whose cardinality scales with model complexity (i.e. number of components and their dimensionalities). We therefore also integrate out the precisions using Gamma priors, $P(\boldsymbol{\nu}|a, b)$, which are conjugate. We use $\boldsymbol{\theta} = \{\Lambda_s, \boldsymbol{\pi}, \boldsymbol{\nu}\}$ to denote model parameters.

Having defined the model and the priors, the variational EM algorithm falls out of Theorem 1. Here we do not provide any details other than to say that the VE step involves computing posteriors over the hidden states in the usual way, and the VM step updates the posteriors over the parameters $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, which have the same form as the priors. We also employ heuristics to search over the model structure space by com-

<table>
<tr><th rowspan="2">number of points per cluster</th><th colspan="6">intrinsic dimensionalities</th></tr>
<tr><th>1</th><th>7</th><th>4</th><th>3</th><th>2</th><th>2</th></tr>
<tr><td>8</td><td colspan="4">2</td><td colspan="2">1</td></tr>
<tr><td>8</td><td colspan="2">1</td><td colspan="4">2</td></tr>
<tr><td>16</td><td>1</td><td colspan="4">4</td><td>2</td></tr>
<tr><td>32</td><td>1</td><td>6</td><td>3</td><td>3</td><td>2</td><td>2</td></tr>
<tr><td>64</td><td>1</td><td>7</td><td>4</td><td>3</td><td>2</td><td>2</td></tr>
<tr><td>128</td><td>1</td><td>7</td><td>4</td><td>3</td><td>2</td><td>2</td></tr>
</table>

Figure 2
Table with learned number of Gaussians and dimensionalities as training set size increases. Boxes represent model components that capture several of the clusters.

paring the evidence lower bounds $\mathcal{F}$ for different structures. Details can be found in [8].

Experiment: Learning MFA model structure. We present just a simple example here to show that in a synthetic problem the variational algorithm can recover both the number of clusters and their intrinsic dimensionalities. We generated a synthetic data set with 300 data points in each of 6 Gaussians with intrinsic dimensionalities (7 4 3 2 2 1) embedded in 10 dimensions. The variational Bayesian approach correctly inferred both the number of Gaussians and their intrinsic dimensionalities. We varied the number of data points and found that, as expected, with fewer points the data could not provide evidence for as many components and intrinsic dimensions (Figure 2).

State-space models

We turn our attention to deriving a variational Bayesian treatment of linear-Gaussian state-space models. This serves two purposes. First, it will illustrate another application of Theorem 1 and an application of Corollary 1. Second, linear-Gaussian state-space models are the cornerstone of stochastic filtering, prediction and control. A variational Bayesian treatment of these models provides a novel way to learn their structure, i.e. to identify the optimal dimensionality of their state-space.

In state-space models (SSMs), a sequence of p-dimensional real-valued observation vectors $(\mathbf{y}_1, \ldots, \mathbf{y}_T)$, denoted $\mathbf{y}_{1:T}$, is modeled by assuming that at each time step t, $\mathbf{y}_t$ was generated from a k-dimensional

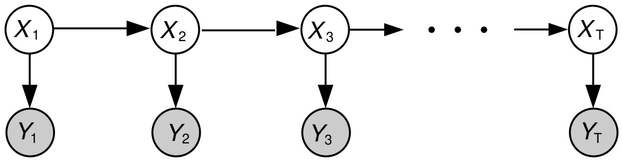

Figure 3
Bayesian network representation of a state-space model.

real-valued hidden state variable $\mathbf{x}_t$, and that the sequence of $\mathbf{x}$'s define a first-order Markov process. The joint probability of a sequence of states and observations is therefore given by:

$$P(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = P(\mathbf{x}_1)P(\mathbf{y}_1|\mathbf{x}_1)\prod_{t=2}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1})P(\mathbf{y}_t|\mathbf{x}_t), \tag{16}$$

This factorization of the joint probability can be represented by the Bayesian network shown in Figure 3.

We focus on models where both the dynamics and output functions are linear and time-invariant and the distribution of the state and observation noise variables is Gaussian, i.e. linear-Gaussian state-space models:

$$\begin{aligned} \mathbf{x}_t &= A\mathbf{x}_{t-1} + \mathbf{w}_t \\ \mathbf{y}_t &= C\mathbf{x}_t + \mathbf{v}_t \end{aligned} \tag{17}$$

where A is the state dynamics matrix and C is the observation matrix. Linear-Gaussian state-space models can be thought of as factor analysis where the factor vector one time step depends linearly on the factor vector at the previous time step. The dynamics can also depend on a driving input $\mathbf{u}_t$:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t + \mathbf{w}_t. \tag{18}$$

Without loss of generality we can assume that $\mathbf{w}_t$ has covariance equal to the unit matrix. The remaining parameters of a linear-Gaussian state-space model with no inputs[5] are the matrices A and C and the covariance matrix of the output noise, $\mathbf{v}_t$, which we will call R and assume to be diagonal, $R = \mathrm{diag}(\rho)^{-1}$, where ρ_i are the *precisions* (inverse variances) associated with each output.

5 It is straightforward to extend the following derivations to SSMs with inputs.

The complete data likelihood for state-space models is Gaussian, which is in the class of exponential family distributions. In order to derive a variational Bayesian algorithm by applying the results in the previous sections we now turn to defining conjugate priors over the parameters.

Each row vector of the A matrix, denoted $\mathbf{a}_i^\top$, is given a zero mean Gaussian prior with inverse covariance matrix equal to diag($\boldsymbol{\alpha}$). Each row vector of C, $\mathbf{c}_i^\top$, is given a zero-mean Gaussian prior with precision matrix equal to diag($\rho_i \boldsymbol{\beta}$). The dependence of the precision of $\mathbf{c}_i^\top$ on the noise output precision ρ_i is motivated by conjugacy. Intuitively, this prior links the scale of the signal and noise.

The prior over the output noise covariance matrix, R, is defined through the precision vector, ρ, which for conjugacy is assumed to be Gamma distributed[6] with hyperparameters a and b:

$$P(\rho\,|a,b) = \prod_{i=1}^{p} \frac{b^a}{\Gamma(a)} \rho_i^{a-1} \exp\{-b\rho_i\}$$

Here, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ are hyperparameters that we can optimise to do automatic relevance determination (ARD) of hidden states, thus inferring the structure of the SSM.

Since A, C, ρ and $\mathbf{x}_{1:T}$ are all unknown, given a sequence of observations $\mathbf{y}_{1:T}$, an exact Bayesian treatment of SSMs would require computing marginals of the posterior $P(A, C, \rho, \mathbf{x}_{1:T}|\mathbf{y}_{1:T})$. This posterior contains interaction terms up to *fifth order* (for example, between elements of C, $\mathbf{x}$ and ρ), and is not analytically manageable. However, since the model is conjugate-exponential we can apply Theorem 1 to derive a variational EM algorithm for state-space models analogous to the maximum-likelihood EM algorithm [33].

Writing out the expression for $\ln P(A, C, \rho, \mathbf{x}_{1:T}, \mathbf{y}_{1:T})$, one sees that it contains interaction terms between ρ and C, but none between A and either ρ or C. This observation implies a further factorisation, $Q(A, C, \rho) = Q(A)Q(C, \rho)$, which falls out of the initial factorisation and the conditional independencies of the model.

Starting from some arbitrary distribution over the hidden variables, the VM step obtained by applying Theorem 1 computes the expected natural parameters of $Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (A, C, \rho)$.

---

6 More generally, if we let R be a full covariance matrix for conjugacy we would give its inverse $V = R^{-1}$ a Wishart distribution: $P(V|\nu, S) \propto |V|^{(\nu-p-1)/2} \exp\left\{-\frac{1}{2}\mathrm{tr}\, VS^{-1}\right\}$, where tr is the matrix trace operator.

We proceed to solve for $Q(A)$. We know from Theorem 1 that $Q(A)$ is multivariate Gaussian, like the prior, so we only need to compute its mean and covariance. A has mean $S^\top(\mathrm{diag}(\boldsymbol{\alpha})+W)^{-1}$ and each row of A has covariance $(\mathrm{diag}(\boldsymbol{\alpha})+W)^{-1}$, where $S=\sum_{t=2}^{T}\left\langle \mathbf{x}_{t-1}\mathbf{x}_t^\top\right\rangle$, $W=\sum_{t=1}^{T-1}\left\langle \mathbf{x}_t\mathbf{x}_t^\top\right\rangle$, and $\langle . \rangle$ denotes averaging w.r.t. the $Q(\mathbf{x}_{1:T})$ distribution.

$Q(C,\rho)$ is also of the same form as the prior. $Q(\rho)$ is a product of Gamma densities $Q(\rho_i)=\mathcal{G}(\rho_i;\tilde{a},\tilde{b_i})$ where $\tilde{a}=a+\frac{T}{2}$, $\tilde{b}=b+\frac{1}{2}g_i$, $g_i=\sum_{t=1}^{T}y_{ti}^2-U_i(\mathrm{diag}(\boldsymbol{\beta})+W')^{-1}U_i^\top$, $U_i=\sum_{t=1}^{T}y_{ti}\langle\mathbf{x}_t^\top\rangle$ and $W'=W+\left\langle \mathbf{x}_T\mathbf{x}_T^\top\right\rangle$. Given ρ, each row of C is Gaussian with covariance $\mathrm{Cov}(\mathbf{c}_i)=(\mathrm{diag}(\boldsymbol{\beta})+W')^{-1}/\rho_i$ and mean $\bar{\mathbf{c}}_i=\rho_i\ U_i\ \mathrm{Cov}(\mathbf{c}_i)$. Note that S, W and U_i are the expected complete data sufficient statistics $\overline{\mathbf{u}}$ mentioned in Theorem 1(a).

We now turn to the VE step: computing $Q(\mathbf{x}_{1:T})$. Since SSMs are singly connected belief networks Corollary 1 tells us that we can make use of belief propagation, which in the case of SSMs is known as the *Kalman smoother* [29]. We therefore run the Kalman smoother with every appearance of the natural parameters of the model replaced with the following corresponding expectations under the Qdistribution: $\langle\rho_i\mathbf{c}_i\rangle$, $\langle\rho_i\mathbf{c}_i\mathbf{c}_i^\top\rangle$, $\langle A\rangle$, $\langle A^\top A\rangle$. We omit the details here. Results from this model are presented in [7].

Other models

Variational Bayesian methods have been applied to several other models, which we mention here briefly. One of the first such models was the mixture of experts architecture [36]. This paper showed that the Q distributions could be optimised in free form. However, because of the softmax gating network in this model, the complete-data likelihood is not exponential so some additional approximations were necessary. In [23] variational methods are applied to hidden Markov models with discrete outputs. These models are conjugate-exponential and furthmore this paper showed that the forward–backward propagation algorithm could be employed (which follows from Corollary 1). A variational Bayesian treatment of probabilistic PCA is given by [3]. Here ARD priors are used to find the optimal dimensionality of the principal component space. Attias [1] shows how the variational Bayesian framework can be applied to mixtures of Gaussians and to a form of independent components analysis (ICA). Since ICA is not conjugate–exponential, a direct variational treatment is not straightforward. However, Attias approximates the ICA model using mixture of Gaussian source distributions, which makes the model conjugate–exponential.

We are currently exploring the boundary of applicability of variational Bayesian methods. In particular Naonori Ueda and the first author have derived variational Bayesian treatments of a conjugate–exponential form of the mixture of experts and hidden Markov model with real-valued outputs. Importantly, much emphasis has been placed on using $\mathcal{F}$ to search over model classes and to avoid local minima in the optimisation. Specifically, using $\mathcal{F}$ it is possible to compare models with different state-space sizes and structures and to incrementally grow or prune structures. This programme has led to models that adapt their structure to the data.

A promising model we plan to explore is the switching state space model, which was analysed in a variational (but non-Bayesian) way in [10]. This is a conjugate–exponential belief network and so is amenable to a variational Bayesian treatment. In fact, this model can be seen as a hybrid between hidden Markov models and state-space models. One amazing property of switching state-space models is that, when coupled with the ability to learn model structure, it is capable of becoming a mixture of factor analysers, mixture of Gaussians, hidden Markov model, or linear dynamical system. So in principle one could run the VB switching SSM model and let it discover the appropriate model class by searching over its possible structures.

8 Sampling from Variational Approximations

One of the limitations of the variational approach is that it only provides a lower bound on the log evidence. While it is possible in certain special cases to form a useful upper bound as well, these bounds are not as generally applicable as the lower bounds. We briefly show how by combining sampling with variational appraches it is possible to estimate the log evidence.

We use one of the least sophisticated sampling techniques: *importance sampling*. In importance sampling, we wish to estimate an expectation of interest under the true distribution $\langle f(x)\rangle_P = \int dx f(x)P(x)$. For some reason this integral is difficult (e.g. it is computationally intractable) and we cannot sample from $P(x)$ although we can evaluate $P(x)$ at any x (perhaps upto a constant). We take n samples $x_i \sim Q(x)$ from a tractable distribution, which has support everywhere $P(x)$ does, and form the estimate

$$\langle f(x) \rangle_P = \int dx\, Q(x) f(x) \frac{P(x)}{Q(x)} \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i) \left[\frac{P(x_i)}{Q(x_i)} \right] \tag{19}$$

The bracketed term is the importance weight w_i.

By importance sampling from the variational approximation we can obtain estimates of three important quantities: the exact predictive density, the true log evidence $\mathcal{L}$, and the KL divergence between the variational posterior and the true posterior. We sample $\boldsymbol{\theta}_i \sim Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$. Each such sample is an instance of our model with predictive density $P(\mathbf{y}|\boldsymbol{\theta}_i)$. We weight these predictive densities by the importance weights $w_i = P(\boldsymbol{\theta}_i, Y)/Q(\boldsymbol{\theta}_i)$, which are easy to evaluate. This results in a *mixture* of models, and will converge to the exact predictive density, $P(\mathbf{y}|Y)$, as long as $Q(\boldsymbol{\theta}) > 0$ wherever $P(\boldsymbol{\theta}|Y) > 0$. The true evidence can be similarly estimated by $P(Y) = \langle w \rangle_Q$, where $\langle \cdot \rangle_Q$ denotes averaging over the importance samples. Finally, the KL divergence is estimated by: $KL(Q(\boldsymbol{\theta}) \| P(\boldsymbol{\theta}|Y)) = \ln\langle w \rangle - \langle \ln w \rangle$.

This procedure has three significant properties. First, the same importance weights can be used to estimate all three quantities. Second, while importance sampling can work very poorly in high dimensions for ad hoc proposal distributions, here the variational optimisation is used in a principled manner to pick Q to be a good approximation to P and therefore hopefully a good proposal distribution. Third, this procedure can be applied to any variational approximation.

Unfortunately, importance sampling is notoriously bad in high dimensions. In fact it is also easy to show that importance sampling can fail even in one dimension (David MacKay, personal communication). Consider computing expectations under a one dimensional Gaussian P by sampling from another Gaussian Q. Although importance sampling can give us unbiased estimates, if the variance of Q is less than half the variance of P the *variance* of the importance weights will be infinite! This problem is exacerbated in higher dimensions, where a mismatch in the tails of P and Q along any dimension could cause similar catastrophic behaviour. There is obviously a great deal of further research that could be put into interesting combinations of sampling methods and variational approximations.

9 Conclusion

Mean field theory and its generalisation in the form of variational methods have provided powerful tools for inference in graphical models. In this chapter we discussed the application of variational methods both in the more traditional maximum-likelihood setting, where it can form the basis of the E step of the EM learning algorithm, and in the Bayesian setting.

In the Bayesian setting variational methods make it possible to lower bound the evidence, which in turn can be used both for model averaging (which we did not discuss here) and model selection. For models in the conjugate–exponential class, the variational Bayesian optimisation turns out to be a generalisation of the EM algorithm. Moreover, propagation algorithms from the graphical model literature can be exploited with (almost) no modification required. These properties should make it possible to automate the derivation of variational Bayesian learning procedures for a large family of models much in the same way as Gibbs sampling and propagation algorithms have been automated in the BUGS [34] and HUGIN [32] software systems, respectively. Through combining sampling, exact propagation algorithms, and variational methods, Bayesian inference in very large domains should be possible, opening up new uses for machine learning, artificial intelligence, and pattern recognition systems.

Acknowledgments

We thank Hagai Attias, Chris Bishop, David MacKay and Naonori Ueda for stimulating discussions and collaborations. This research was funded by the Gatsby Charitable Foundation.

References

[1]H. Attias. A variational bayesian framework for graphical models. In *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge, MA, 2000.

[2]L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41:164–171, 1970.

[3]C.M. Bishop. Variational PCA. In *Proc. Ninth Int. Conf. on Artificial Neural Networks. ICANN*, 1999.

[4]R. Cox. Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946.

[5]A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B*, 39:1–38, 1977.

[6]Z. Ghahramani. Factorial learning and the *EM* algorithm. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 617–624. MIT Press, Cambridge, MA, 1995.

[7]Z. Ghahramani and M. J. Beal. Propagation algorithms for variational bayesian learning. In *Submitted*. 2000.

[8]Z. Ghahramani and M.J. Beal. Variational inference for Bayesian mixtures of factor analysers. In *Adv. Neur. Inf. Proc. Sys. 12*. MIT Press, 2000.

[9]Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1 [http://www.gatsby.ucl.ac.uk/∼zoubin/papers/tr-96-1.ps.gz], Department of Computer Science, University of Toronto, 1996.

[10]Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4), 2000.

[11]Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–273, 1997.

[12]D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06 [ftp://ftp.research.microsoft.com/pub/tr/TR-95-06.PS] , Microsoft Research, 1996.

[13]G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length, and Helmholtz free energy. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann Publishers, San Francisco, CA, 1994.

[14]G.E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Sixth ACM Conference on Computational Learning Theory, Santa Cruz*, 1993.

[15]T. S. Jaakkola. Variational methods for Inference and estimation in graphical models. Technical Report Ph.D. Thesis, Cambridge, MA, 1997.

[16]E.T. Jaynes. *Probability Theory: The Logic of Science*. 1995.

[17]W.H. Jefferys and J.O. Berger. Ockham's razor and Bayesian analysis. *American Scientist*, 80:64–72, 1992.

[18]F. V. Jensen. *Introduction to Bayesian Networks*. Springer-Verlag, New York, 1996.

[19]M.I. Jordan, editor. *Learning in Graphical Models*. Kluwer Academic Press. Also available from MIT Press (paperback)., 1998.

[20]M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K Saul. An introduction to variational methods in graphical models. *Machine Learning*, 37:183–233, 1999.

[21]S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. Royal Statistical Society B*, pages 157–224, 1988.

[22]D. J. C. MacKay. Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505, 1995.

[23]D.J.C. MacKay. Ensemble learning for hidden Markov models. Technical

report, Cavendish Laboratory, University of Cambridge, 1997.

[24]R. M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113, 1992.

[25]R. M. Neal. Probabilistic inference using Markov chain monte carlo methods. Technical Report CRG-TR-93-1, 1993.

[26]R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic Press, 1998.

[27]R.M. Neal. Assessing relevance determination methods using DELVE. In C.M. Bishop, editor, *Neural Networks and Machine Learning*, pages 97–129. Springer-Verlag, 1998.

[28]J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[29]H. E. Rauch. Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8:371–372, 1963.

[30]S. T. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[31]L.K. Saul, T. Jaakkola, and M. I. Jordan. Mean Field Theory for Sigmoid Belief Networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

[32]see www.hugin.dk .

[33]R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *J. Time Series Analysis*, 3(4):253–264, 1982.

[34]D. J. Spiegelhalter, A Thomas, and N G Best. Computation on bayesian graphical models. *Bayesian Statistics*, 5:407–425 (see www.mrc–bsu.cam.ac.uk/bugs), 1996.

[35]M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[36]S. Waterhouse, D.J.C. Mackay, and T. Robinson. Bayesian methods for mixtures of experts. In *Adv. Neur. Inf. Proc. Sys. 7*. MIT Press, 1995.

[37]C. K. I. Williams and G. E. Hinton. Mean field networks that learn to discriminate temporally distorted strings. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski, and G.E. Hinton, editors, *Connectionist Models: Proceedings of the 1990 Summer School*, pages 18–22. Morgan Kaufmann Publishers, San Mateo, CA, 1991.