Machine Learning, ?, 1–31 (1997)

Factorial Hidden Markov Models

ZOUBIN GHAHRAMANI zoubin@cs.toronto.edu
Department of Computer Science, University of Toronto, Toronto, ON M5S 3H5, Canada

MICHAEL I. JORDAN jordan@psyche.mit.edu
Department of Brain & Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Editor: Padhraic Smyth

Abstract. Hidden Markov models (HMMs) have proven to be one of the most widely used tools for learning probabilistic models of time series data. In an HMM, information about the past is conveyed through a single discrete variable—the hidden state. We discuss a generalization of HMMs in which this state is factored into multiple state variables and is therefore represented in a distributed manner. We describe an exact algorithm for inferring the posterior probabilities of the hidden state variables given the observations, and relate it to the forward–backward algorithm for HMMs and to algorithms for more general graphical models. Due to the combinatorial nature of the hidden state representation, this exact algorithm is intractable. As in other intractable systems, approximate inference can be carried out using Gibbs sampling or variational methods. Within the variational framework, we present a structured approximation in which the the state variables are decoupled, yielding a tractable algorithm for learning the parameters of the model. Empirical comparisons suggest that these approximations are efficient and provide accurate alternatives to the exact methods. Finally, we use the structured approximation to model Bach's chorales and show that factorial HMMs can capture statistical structure in this data set which an unconstrained HMM cannot.

Keywords: Hidden Markov models, time series, EM algorithm, graphical models, Bayesian networks, mean field theory

1. Introduction

Due to its flexibility and to the simplicity and efficiency of its parameter estimation algorithm, the hidden Markov model (HMM) has emerged as one of the basic statistical tools for modeling discrete time series, finding widespread application in the areas of speech recognition (Rabiner & Juang, 1986) and computational molecular biology (Krogh, Brown, Mian, Sjölander, & Haussler, 1994). An HMM is essentially a mixture model, encoding information about the history of a time series in the value of a single multinomial variable—the hidden state—which can take on one of K discrete values. This multinomial assumption supports an efficient parameter estimation algorithm—the Baum-Welch algorithm—which considers each of the K settings of the hidden state at each time step. However, the multinomial assumption also severely limits the representational capacity of HMMs. For example, to represent 30 bits of information about the history of a time sequence, an HMM would need $K = 2^{30}$ distinct states. On the other hand, an HMM with a *distributed* state representation could achieve the same task with 30 binary state variables (Williams & Hinton, 1991). This paper addresses the problem of constructing efficient learning algorithms for hidden Markov models with distributed state representations.

The need for distributed state representations in HMMs can be motivated in two ways. First, such representations let the model automatically decompose the state space into features that decouple the dynamics of the process that generated the data. Second, distributed state representations simplify the task of modeling time series that are known a priori to be generated from an interaction of multiple, loosely-coupled processes. For example, a speech signal generated by the superposition of multiple simultaneous speakers can be potentially modeled with such an architecture.

Williams and Hinton (1991) first formulated the problem of learning in HMMs with distributed state representations and proposed a solution based on deterministic Boltzmann learning.[1] The approach presented in this paper is similar to Williams and Hinton's in that it can also be viewed from the framework of statistical mechanics and mean field theory. However, our learning algorithm is quite different in that it makes use of the special structure of HMMs with a distributed state representation, resulting in a significantly more efficient learning procedure. Anticipating the results in Section 3, this learning algorithm obviates the need for the two-phase procedure of Boltzmann machines, has an exact M step, and makes use of the forward–backward algorithm from classical HMMs as a subroutine. A different approach comes from Saul and Jordan (1995), who derived a set of rules for computing the gradients required for learning in HMMs with distributed state spaces. However, their methods can only be applied to a limited class of architectures.

Hidden Markov models with distributed state representations are a particular class of probabilistic graphical model (Pearl, 1988; Lauritzen & Spiegelhalter, 1988), which represent probability distributions as graphs in which the nodes correspond to random variables and the links represent conditional independence relations. The relation between hidden Markov models and graphical models has recently been reviewed in Smyth, Heckerman and Jordan (1997). Although exact probability propagation algorithms exist for general graphical models (Jensen, Lauritzen, & Olesen, 1990), these algorithms are intractable for densely-connected models such as the ones we consider in this paper. One approach to dealing with this issue is to utilize stochastic sampling methods (Kanazawa et al., 1995). Another approach, which provides the basis for algorithms described in the current paper, is to make use of variational methods (cf. Saul, Jaakkola, & Jordan, 1996).

In the following section we define the probabilistic model for factorial HMMs and in Section 3 we present algorithms for inference and learning. In Section 4 we describe empirical results comparing exact and approximate algorithms for learning on the basis of time complexity and model quality. We also apply factorial HMMs to a real time series data set consisting of the melody lines from a collection of chorales by J. S. Bach. We discuss several generalizations of the probabilistic model

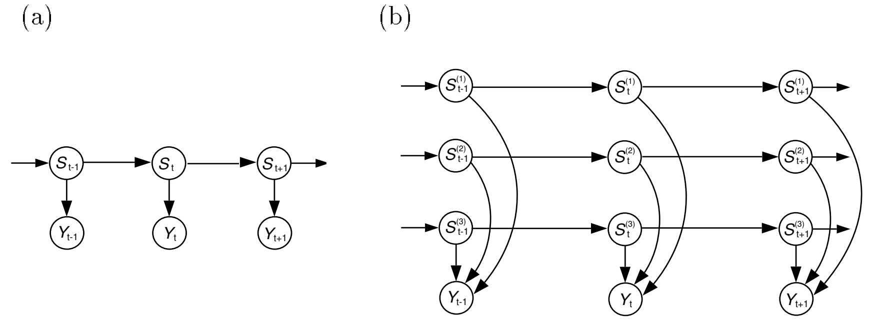

Figure 1. (a) A directed acyclic graph (DAG) specifying conditional independence relations for a hidden Markov model. Each node is conditionally independent from its non-descendants given its parents. (b) A DAG representing the conditional independence relations in a factorial HMM with $M = 3$ underlying Markov chains.

in Section 5, and we conclude in Section 6. Where necessary, details of derivations are provided in the appendixes.

2. The probabilistic model

We begin by describing the hidden Markov model, in which a sequence of observations $\{Y_t\}$ where $t = 1, \ldots T$, is modeled by specifying a probabilistic relation between the observations and a sequence of hidden states $\{S_t\}$, and a Markov transition structure linking the hidden states. The model assumes two sets of conditional independence relations: that Y_t is independent of all other observations and states given S_t, and that S_t is independent of $S_1 \ldots S_{t-2}$ given S_{t-1} (the first-order Markov property). Using these independence relations, the joint probability for the sequence of states and observations can be factored as

$$P(\{S_t, Y_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t). \tag{1}$$

The conditional independencies specified by equation (1) can be expressed graphically in the form of Figure 1 (a). The state is a single multinomial random variable that can take one of K discrete values, $S_t \in \{1, \ldots, K\}$. The state transition probabilities, $P(S_t|S_{t-1})$, are specified by a $K \times K$ transition matrix. If the observations are discrete symbols taking on one of D values, the observation probabilities $P(Y_t|S_t)$ can be fully specified as a $K \times D$ observation matrix. For a continuous observation vector, $P(Y_t|S_t)$ can be modeled in many different forms, such as a Gaussian, a mixture of Gaussians, or even a neural network.[2]

In the present paper, we generalize the HMM state representation by letting the state be represented by a collection of state variables

$$S_t = S_t^{(1)}, \ldots S_t^{(m)}, \ldots, S_t^{(M)}, \tag{2}$$

each of which can take on $K^{(m)}$ values. We refer to these models as *factorial hidden Markov models*, as the state space consists of the cross product of these state variables. For simplicity, we will assume that $K^{(m)} = K$, for all m, although all the results we present can be trivially generalized to the case of differing $K^{(m)}$. Given that the state space of this factorial HMM consists of all K^M combinations of the $S_t^{(m)}$ variables, placing no constraints on the state transition structure would result in a $K^M \times K^M$ transition matrix. Such an unconstrained system is uninteresting for several reasons: it is equivalent to an HMM with K^M states; it is unlikely to discover any interesting structure in the K state variables, as all variables are allowed to interact arbitrarily; and both the time complexity and sample complexity of the estimation algorithm are exponential in M.

We therefore focus on factorial HMMs in which the underlying state transitions are constrained. A natural structure to consider is one in which each state variable evolves according to its own dynamics, and is *a priori* uncoupled from the other state variables:

$$P(S_t|S_{t-1}) = \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}). \tag{3}$$

A graphical representation for this model is presented in Figure 1 (b). The transition structure for this system can be represented as M distinct $K \times K$ matrices. Generalizations that allow coupling between the state variables are briefly discussed in Section 5.

As shown in Figure 1 (b), in a factorial HMM the observation at time step t can depend on all the state variables at that time step. For continuous observations, one simple form for this dependence is linear Gaussian; that is, the observation Y_t is a Gaussian random vector whose mean is a linear function of the state variables. We represent the state variables as $K \times 1$ vectors, where each of the K discrete values corresponds to a 1 in one position and 0 elsewhere. The resulting probability density for a $D \times 1$ observation vector Y_t is

$$P(Y_t|S_t) = |C|^{-1/2}\,(2\pi)^{-D/2}\,\exp\left\{-\frac{1}{2}\,(Y_t - \mu_t)'\,C^{-1}\,(Y_t - \mu_t)\right\}, \tag{4a}$$

where

$$\mu_t = \sum_{m=1}^{M} W^{(m)} S_t^{(m)}. \tag{4b}$$

Each $W^{(m)}$ matrix is a $D \times K$ matrix whose columns are the contributions to the means for each of the settings of $S_t^{(m)}$, C is the $D \times D$ covariance matrix, $'$ denotes matrix transpose, and $|\cdot|$ is the matrix determinant operator.

One way to understand the observation model in equations (4a) and (4b) is to consider the marginal distribution for Y_t, obtained by summing over the possible states. There are K settings for each of the M state variables, and thus there are K^M possible mean vectors obtained by forming sums of M columns where one column is chosen from each of the $W^{(m)}$ matrices. The resulting marginal density of Y_t is thus a Gaussian mixture model, with K^M Gaussian mixture components each having a constant covariance matrix C. This static mixture model, without inclusion of the time index and the Markov dynamics, is a factorial parameterization of the standard mixture of Gaussians model that has interest in its own right (Zemel, 1993; Hinton & Zemel, 1994; Ghahramani, 1995). The model we are considering in the current paper extends this model by allowing Markov dynamics in the discrete state variables underlying the mixture. Unless otherwise stated, we will assume the Gaussian observation model throughout the paper.

The hidden state variables at one time step, although marginally independent, become conditionally dependent given the observation sequence. This can be determined by applying the semantics of directed graphs, in particular the d-separation criterion (Pearl, 1988), to the graphical model in Figure 1 (b). Consider the Gaussian model in equations (4a)-(4b). Given an observation vector Y_t, the posterior probability of each of the settings of the hidden state variables is proportional to the probability of Y_t under a Gaussian with mean μ_t. Since μ_t is a function of all the state variables, the probability of a setting of one of the state variables will depend on the setting of the other state variables.[3] This dependency effectively couples all of the hidden state variables for the purposes of calculating posterior probabilities and makes exact inference intractable for the factorial HMM.

3. Inference and learning

The inference problem in a probabilistic graphical model consists of computing the probabilities of the hidden variables given the observations. In the context of speech recognition, for example, the observations may be acoustic vectors and the goal of inference may be to compute the probability for a particular word or sequence of phonemes (the hidden state). This problem can be solved efficiently via the forward–backward algorithm (Rabiner & Juang, 1986), which can be shown to be a special case of the Jensen, Lauritzen, and Olesen (1990) algorithm for probability propagation in more general graphical models (Smyth et al., 1997). In some cases, rather than a probability distribution over hidden states it is desirable to infer the single most probable hidden state sequence. This can be achieved via the Viterbi (1967) algorithm, a form of dynamic programming that is very closely related to the forward–backward algorithm and also has analogues in the graphical model literature (Dawid, 1992).

The learning problem for probabilistic models consists of two components: learning the structure of the model and learning its parameters. Structure learning is a topic of current research in both the graphical model and machine learning communities (e.g. Heckerman, 1995; Stolcke & Omohundro, 1993). In the current paper we deal exclusively with the problem of learning the parameters for a given structure.

3.1. The EM algorithm

The parameters of a factorial HMM can be estimated via the expectation maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977), which in the case of classical HMMs is known as the Baum–Welch algorithm (Baum, Petrie, Soules, & Weiss, 1970). This procedure iterates between a step that fixes the current parameters and computes posterior probabilities over the hidden states (the E step) and a step that uses these probabilities to maximize the expected log likelihood of the observations as a function of the parameters (the M step). Since the E step of EM is exactly the inference problem as described above, we subsume the discussion of both inference and learning problems into our description of the EM algorithm for factorial HMMs.

The EM algorithm follows from the definition of the expected log likelihood of the complete (observed and hidden) data:

$$\mathcal{Q}(\phi^{\text{new}}|\phi) = E\left\{ \log P(\{S_t, Y_t\}|\phi^{\text{new}}) \mid \phi, \{Y_t\}\right\}, \tag{5}$$

where $\mathcal{Q}$ is a function of the parameters ϕ^{new}, given the current parameter estimate ϕ and the observation sequence $\{Y_t\}$. For the factorial HMM the parameters of the model are $\phi = \{W^{(m)}, \pi^{(m)}, P^{(m)}, C\}$, where $\pi^{(m)} = P(S_1^{(m)})$ and $P^{(m)} = P(S_t^{(m)}|S_{t-1}^{(m)})$. The E step consists of computing $\mathcal{Q}$. By expanding (5) using equations (1)–(4b), we find that $\mathcal{Q}$ can be expressed as a function of three types of expectations over the hidden state variables: $\langle S_t^{(m)}\rangle$, $\langle S_t^{(m)} S_t^{(n)'}\rangle$, and $\langle S_{t-1}^{(m)} S_t^{(m)'}\rangle$, where $\langle\cdot\rangle$ has been used to abbreviate $E\{\cdot|\phi,\{Y_t\}\}$. In the HMM notation of Rabiner and Juang (1986), $\langle S_t^{(m)}\rangle$ corresponds to γ_t, the vector of state occupation probabilities, $\langle S_{t-1}^{(m)} S_t^{(m)'}\rangle$ corresponds to ξ_t, the $K \times K$ matrix of state occupation probabilities at two consecutive time steps, and $\langle S_t^{(m)} S_t^{(n)'}\rangle$ has no analogue when there is only a single underlying Markov model. The M step uses these expectations to maximize $\mathcal{Q}$ as a function of ϕ^{new}. Using Jensen's inequality, Baum, Petrie, Soules & Weiss (1970) showed that each iteration of the E and M steps increases the likelihood, $P(\{Y_t\}|\phi)$, until convergence to a (local) optimum.

As in hidden Markov models, the exact M step for factorial HMMs is simple and tractable. In particular, the M step for the parameters of the output model described in equations (4a)-(4b) can be found by solving a weighted linear regression problem. Similarly, the M steps for the priors, $\pi^{(m)}$, and state transition matrices, $P^{(m)}$, are identical to the ones used in the Baum–Welch algorithm. The details of the M step are given in Appendix A. We now turn to the substantially more difficult problem of computing the expectations required for the E step.

3.2. Exact inference

Unfortunately, the exact E step for factorial HMMs is computationally intractable. This fact can best be shown by making reference to standard algorithms for probabilistic inference in graphical models (Lauritzen & Spiegelhalter, 1988), although it can also be derived readily from direct application of Bayes rule. Consider the computations that are required for calculating posterior probabilities for the factorial HMM shown in Figure 1 (b) within the framework of the Lauritzen and Spiegelhalter algorithm. Moralizing and triangulating the graphical structure for the factorial HMM results in a junction tree (in fact a chain) with $T(M+1)-M$ cliques of size $M+1$. The resulting probability propagation algorithm has time complexity $O(TMK^{M+1})$ for a single observation sequence of length T. We present a forward–backward type recursion that implements the exact E step in Appendix B. The naive exact algorithm which consists of translating the factorial HMM into an equivalent HMM with K^M states and using the forward–backward algorithm, has time complexity $O(TK^{2M})$. Like other models with multiple densely-connected hidden variables, this exponential time complexity makes exact learning and inference intractable.

Thus, although the Markov property can be used to obtain forward–backward-like factorizations of the expectations across time steps, the sum over all possible configurations of the other hidden state variables *within* each time step is unavoidable. This intractability is due inherently to the cooperative nature of the model: for the Gaussian output model, for example, the settings of all the state variables at one time step cooperate in determining the mean of the observation vector.

3.3. Inference using Gibbs sampling

Rather than computing the exact posterior probabilities, one can approximate them using a Monte Carlo sampling procedure, and thereby avoid the sum over exponentially many state patterns at some cost in accuracy. Although there are many possible sampling schemes (for a review see Neal, 1993), here we present one of the simplest—Gibbs sampling (Geman & Geman, 1984). For a given observation sequence $\{Y_t\}$, this procedure starts with a random setting of the hidden states $\{S_t\}$. At each step of the sampling process, each state vector is updated stochastically according to its probability distribution conditioned on the setting of all the other state vectors. The graphical model is again useful here, as each node is conditionally independent of all other nodes given its Markov blanket, defined as the set of children, parents, and parents of the children of a node. To sample from a typical state variable $S_t^{(m)}$ we only need to examine the states of a few neighboring nodes:

$$\begin{aligned} S_t^{(m)} \text{ sampled from } & P(S_t^{(m)}|\{S_t^{(n)} : n \neq m\}, S_{t-1}^{(m)}, S_{t+1}^{(m)}, Y_t) \\ \propto \; & P(S_t^{(m)}|S_{t-1}^{(m)})\; P(S_{t+1}^{(m)}|S_t^{(m)})\; P(Y_t|S_t^{(1)}, \ldots, S_t^{(m)}, \ldots, S_t^{(M)}). \end{aligned}$$

Sampling once from each of the TM hidden variables in the model results in a new sample of the hidden state of the model and requires $O(TMK)$ operations. The sequence of overall states resulting from each pass of Gibbs sampling defines a Markov chain over the state space of the model. Assuming that all probabilities are bounded away from zero, this Markov chain is guaranteed to converge to the posterior probabilities of the states given the observations (Geman & Geman, 1984). Thus, after some suitable time, samples from the Markov chain can be taken as approximate samples from the posterior probabilities. The first and second-order statistics needed to estimate $\langle S_t^{(m)} \rangle$, $\langle S_t^{(m)} S_t^{(n)'} \rangle$ and $\langle S_{t-1}^{(m)} S_t^{(m)'} \rangle$ are collected using the states visited and the probabilities estimated during this sampling process are used in the approximate E step of EM.[4]

3.4. Completely factorized variational inference

There also exists a second approximation of the posterior probability of the hidden states that is both tractable and deterministic. The basic idea is to approximate the posterior distribution over the hidden variables $P(\{S_t\}|\{Y_t\})$ by a tractable distribution $Q(\{S_t\})$. This approximation provides a lower bound on the log likelihood that can be used to obtain an efficient learning algorithm.

The argument can be formalized following the reasoning of Saul, Jaakkola, and Jordan (1996). Any distribution over the hidden variables $Q(\{S_t\})$ can be used to define a lower bound on the log likelihood

$$\begin{aligned}
\log P(\{Y_t\}) &= \log \sum_{\{S_t\}} P(\{S_t, Y_t\}) \\
&= \log \sum_{\{S_t\}} Q(\{S_t\}) \left[\frac{P(\{S_t, Y_t\})}{Q(\{S_t\})} \right] \\
&\geq \sum_{\{S_t\}} Q(\{S_t\}) \log \left[\frac{P(\{S_t, Y_t\})}{Q(\{S_t\})} \right],
\end{aligned}$$

where we have made use of Jensen's inequality in the last step. The difference between the left-hand side and the right-hand side of this inequality is given by the Kullback-Leibler divergence (Cover & Thomas, 1991):

$$\mathrm{KL}(Q\|P) = \sum_{\{S_t\}} Q(\{S_t\}) \log \left[\frac{Q(\{S_t\})}{P(\{S_t\}|\{Y_t\})} \right]. \tag{6}$$

The complexity of exact inference in the approximation given by Q is determined by its conditional independence relations, not by its parameters. Thus, we can chose Q to have a tractable structure—a graphical representation that eliminates some of the dependencies in P. Given this structure, we are free to vary the parameters of Q so as to obtain the tightest possible bound by minimizing (6).

We will refer to the general strategy of using a parameterized approximating distribution as a *variational approximation* and refer to the free parameters of the distribution as *variational parameters*. To illustrate, consider the simplest variational approximation, in which the state variables are assumed independent given the observations (Figure 2 (a)). This distribution can be written as

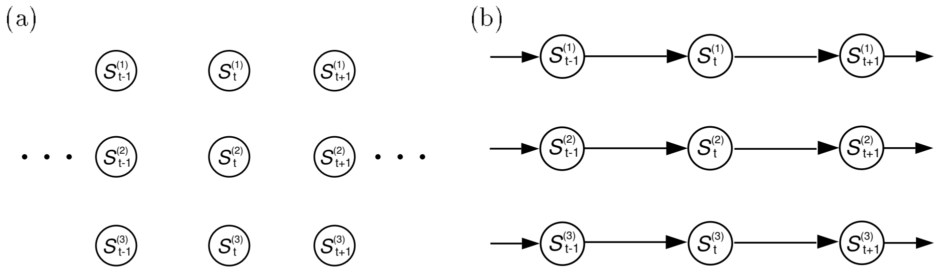

Figure 2. (a) The completely factorized variational approximation assumes that all the state variables are independent (conditional on the observation sequence). (b) The structured variational approximation assumes that the state variables retain their Markov structure within each chain, but are independent across chains.

$$Q(\{S_t\}|\theta) = \prod_{t=1}^{T} \prod_{m=1}^{M} Q(S_t^{(m)}|\theta_t^{(m)}). \tag{7}$$

The variational parameters, $\theta = \{\theta_t^{(m)}\}$, are the means of the state variables, where, as before, a state variable $S_t^{(m)}$ is represented as a K-dimensional vector with a 1 in the k^{th} position and 0 elsewhere, if the m^{th} Markov chain is in state k at time t. The elements of the vector $\theta_t^{(m)}$ therefore define the state occupation probabilities for the multinomial variable $S_t^{(m)}$ under the distribution Q:

$$Q(S_t^{(m)}|\theta_t^{(m)}) = \prod_{k=1}^{K} \left(\theta_{t,k}^{(m)}\right)^{S_{t,k}^{(m)}} \qquad \text{where} \quad S_{t,k}^{(m)} \in \{0,1\}; \quad \sum_{k=1}^{K} S_{t,k}^{(m)} = 1. \tag{8}$$

This completely factorized approximation is often used in statistical physics, where it provides the basis for simple yet powerful *mean field approximations* to statistical mechanical systems (Parisi, 1988).

To make the bound as tight as possible we vary θ separately for each observation sequence so as to minimize the KL divergence. Taking the derivatives of (6) with respect to $\theta_t^{(m)}$ and setting them to zero, we obtain the set of fixed point equations (see Appendix C) defined by

$$\theta_t^{(m)\,\text{new}} = \varphi\left\{W^{(m)\prime} C^{-1} \tilde{Y}_t^{(m)} - \frac{1}{2}\Delta^{(m)} + (\log P^{(m)})\, \theta_{t-1}^{(m)} + (\log P^{(m)})'\, \theta_{t+1}^{(m)}\right\} \tag{9a}$$

where $\tilde{Y}_t^{(m)}$ is the residual error in Y_t given the predictions from all the state variables not including m:

$$\tilde{Y}_t^{(m)} \equiv Y_t - \sum_{\ell \neq m}^{M} W^{(\ell)} \theta_t^{(\ell)}, \tag{9b}$$

$\Delta^{(m)}$ is the vector of diagonal elements of $W^{(m)\prime}C^{-1}W^{(m)}$, and $\varphi\{\cdot\}$ is the softmax operator, which maps a vector A into a vector B of the same size, with elements

$$B_i = \frac{\exp\{A_i\}}{\sum_j \exp\{A_j\}}, \tag{10}$$

and $\log P^{(m)}$ denotes the elementwise logarithm of the transition matrix $P^{(m)}$.

The first term of (9a) is the projection of the error in reconstructing the observation onto the weights of state vector m—the more a particular setting of a state vector can reduce this error, the larger its associated variational parameter. The second term arises from the fact that the second order correlation $\langle S_t^{(m)} S_t^{(m)} \rangle$ evaluated under the variational distribution is a diagonal matrix composed of the elements of $\theta_t^{(m)}$. The last two terms introduce dependencies forward and backward in time.[5] Therefore, although the posterior distribution over the hidden variables is approximated with a completely factorized distribution, the fixed point equations couple the parameters associated with each node with the parameters of its Markov blanket. In this sense, the fixed point equations propagate information along the same pathways as those defining the exact algorithms for probability propagation.

The following may provide an intuitive interpretation of the approximation being made by this distribution. Given a particular observation sequence, the hidden state variables for the M Markov chains at time step t are stochastically coupled. This stochastic coupling is approximated by a system in which the hidden variables are uncorrelated but have coupled means. The variational or "mean-field" equations solve for the deterministic coupling of the means that best approximates the stochastically coupled system.

Each hidden state vector is updated in turn using (9a), with a time complexity of $O(TMK^2)$ per iteration. Convergence is determined by monitoring the KL divergence in the variational distribution between successive time steps; in practice convergence is very rapid (about 2 to 10 iterations of (9a)). Once the fixed point equations have converged, the expectations required for the E step can be obtained as a simple function of the parameters (equations (C.6)–(C.8) in Appendix C).

3.5. Structured variational inference

The approximation presented in the previous section factors the posterior probability such that all the state variables are statistically independent. In contrast to this rather extreme factorization, there exists a third approximation that is both tractable and preserves much of the probabilistic structure of the original system. In this scheme, the factorial HMM is approximated by M uncoupled HMMs as shown in Figure 2 (b). Within each HMM, efficient and exact inference is implemented via the forward–backward algorithm. The approach of exploiting such tractable substructures was first suggested in the machine learning literature by Saul and Jordan (1996).

Note that the arguments presented in the previous section did not hinge on the the form of the approximating distribution. Therefore, more structured variational approximations can be obtained by using more structured variational distributions Q. Each such Q provides a lower bound on the log likelihood and can be used to obtain a learning algorithm.

We write the structured variational approximation as

$$Q(\{S_t\}|\theta) = \frac{1}{Z_Q} \prod_{m=1}^{M} Q(S_1^{(m)}|\theta) \prod_{t=2}^{T} Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta), \tag{11a}$$

where Z_Q is a normalization constant ensuring that Q integrates to one and

$$Q(S_1^{(m)}|\theta) = \prod_{k=1}^{K} \left(h_{1,k}^{(m)} \pi_k^{(m)} \right)^{S_{1,k}^{(m)}} \tag{11b}$$

$$\begin{aligned} Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta) &= \prod_{k=1}^{K} \left(h_{t,k}^{(m)} \sum_{j=1}^{K} P_{k,j}^{(m)} S_{t-1,j}^{(m)} \right)^{S_{t,k}^{(m)}} \\ &= \prod_{k=1}^{K} \left(h_{t,k}^{(m)} \prod_{j=1}^{K} \left(P_{k,j}^{(m)} \right)^{S_{t-1,j}^{(m)}} \right)^{S_{t,k}^{(m)}}, \end{aligned} \tag{11c}$$

where the last equality follows from the fact that $S_{t-1}^{(m)}$ is a vector with a 1 in one position and 0 elsewhere. The parameters of this distribution are $\theta = \{\pi^{(m)}, P^{(m)}, h_t^{(m)}\}$— the original priors and state transition matrices of the factorial HMM and a time-varying bias for each state variable. Comparing equations (11a)–(11c) to equation (1), we can see that the $K \times 1$ vector $h_t^{(m)}$ plays the role of the probability of an observation ($P(Y_t|S_t)$ in (1)) for each of the K settings of $S_t^{(m)}$. For example, $Q(S_{1,j}^{(m)} = 1|\theta) = h_{1,j}^{(m)}\ P(S_{1,j}^{(m)} = 1|\phi)$ corresponds to having an observation at time $t = 1$ that under state $S_{1,j}^{(m)} = 1$ has probability $h_{1,j}^{(m)}$.

Intuitively, this approximation uncouples the M Markov chains and attaches to each state variable a distinct fictitious observation. The probability of this fictitious observation can be varied so as to minimize the KL divergence between Q and P.

Applying the same arguments as before, we obtain a set of fixed point equations for $h_t^{(m)}$ that minimize $\mathrm{KL}(Q\|P)$, as detailed in Appendix D:

$$h_t^{(m)\text{ new}} = \exp\left\{ W^{(m)\prime} C^{-1} \tilde{Y}_t^{(m)} - \frac{1}{2}\Delta^{(m)} \right\}, \tag{12a}$$

where $\Delta^{(m)}$ is defined as before, and where we redefine the residual error to be

$$\tilde{Y}_t^{(m)} \equiv Y_t - \sum_{\ell \neq m}^{M} W^{(\ell)} \langle S_t^{(\ell)} \rangle. \tag{12b}$$

The parameter $h_t^{(m)}$ obtained from these fixed point equations is the observation probability associated with state variable $S_t^{(m)}$ in hidden Markov model m. Using these probabilities, the forward–backward algorithm is used to compute a new set of expectations for $\langle S_t^{(m)} \rangle$, which are fed back into (12a) and (12b). This algorithm is therefore used as a subroutine in the minimization of the KL divergence.

Note the similarity between equations (12a)–(12b) and equations (9a)–(9b) for the completely factorized system. In the completely factorized system, since $\langle S_t^{(m)} \rangle = \theta_t^{(m)}$, the fixed point equations can be written explicitly in terms of the variational parameters. In the structured approximation, the dependence of $\langle S_t^{(m)} \rangle$ on $h_t^{(m)}$ is computed via the forward–backward algorithm. Note also that (12a) does not contain terms involving the prior, $\pi^{(m)}$, or transition matrix, $P^{(m)}$. These terms have cancelled by our choice of approximation.

3.6. Choice of approximation

The theory of the EM algorithm as presented in Dempster et al. (1977) assumes the use of an exact E step. For models in which the exact E step is intractable, one must instead use an approximation like those we have just described. The choice among these approximations must take into account several theoretical and practical issues.

Monte Carlo approximations based on Markov chains, such as Gibbs sampling, offer the theoretical assurance that the sampling procedure will converge to the correct posterior distribution in the limit. Although this means that one can come arbitrarily close to the exact E step, in practice convergence can be slow (especially for multimodal distributions) and it is often very difficult to determine how close one is to convergence. However, when sampling is used for the E step of EM, there is a time tradeoff between the number of samples used and the number of EM iterations. It seems wasteful to wait until convergence early on in learning, when the posterior distribution from which samples are drawn is far from the posterior given the optimal parameters. In practice we have found that even approximate E steps using very few Gibbs samples (e.g. around ten samples of each hidden variable) tend to increase the true likelihood.

Variational approximations offer the theoretical assurance that a lower bound on the likelihood is being maximized. Both the minimization of the KL divergence in the E step and the parameter update in the M step are guaranteed not to decrease this lower bound, and therefore convergence can be defined in terms of the bound. An alternative view given by Neal and Hinton (1993) describes EM in terms of the negative free energy, F, which is a function of the parameters, ϕ, the observations, Y, and a posterior probability distribution over the hidden variables, $Q(S)$:

$$F(Q, \phi) = E_Q \{\log P(Y, S|\phi)\} - E_Q \{\log Q(S)\},$$

where E_Q denotes expectation over S using the distribution $Q(S)$. The exact E step in EM maximizes F with respect to Q given ϕ. The variational E steps used here maximize F with respect to Q given ϕ, subject to the constraint that Q is of a particular tractable form. Given this view, it seems clear that the structured approximation is preferable to the completely factorized approximation since it places fewer constraints on Q, at no cost in tractability.

4. Experimental results

To investigate learning and inference in factorial HMMs we conducted two experiments. The first experiment compared the different approximate and exact methods of inference on the basis of computation time and the likelihood of the model obtained from synthetic data. The second experiment sought to determine whether the decomposition of the state space in factorial HMMs presents any advantage in modeling a real time series data set that we might assume to have complex internal structure—Bach's chorale melodies.

4.1. Experiment 1: Performance and timing benchmarks

Using data generated from a factorial HMM structure with M underlying Markov models with K states each, we compared the time per EM iteration and the training and test set likelihoods of five models:

- HMM trained using the Baum-Welch algorithm;
- Factorial HMM trained with exact inference for the E step, using a straightforward application of the forward–backward algorithm, rather than the more efficient algorithm outlined in Appendix B;
- Factorial HMM trained using Gibbs sampling for the E step with the number of samples fixed at 10 samples per variable;[6]
- Factorial HMM trained using the completely factorized variational approximation; and
- Factorial HMM trained using the structured variational approximation.

All factorial HMMs consisted of M underlying Markov models with K states each, whereas the HMM had K^M states. The data were generated from a factorial HMM structure with M state variables, each of which could take on K discrete values. All of the parameters of this model, except for the output covariance matrix, were sampled from a uniform $[0, 1]$ distribution and appropriately normalized to satisfy the sum-to-one constraints of the transition matrices and priors. The covariance matrix was set to a multiple of the identity matrix $C = 0.0025I$.

The training and test sets consisted of 20 sequences of length 20, where the observable was a four-dimensional vector. For each randomly sampled set of parameters, a separate training set and test set were generated and each algorithm was run once.

Fifteen sets of parameters were generated for each of the four problem sizes. Algorithms were run for a maximum of 100 iterations of EM or until convergence, defined as the iteration k at which the log likelihood $L(k)$, or approximate log likelihood if an approximate algorithm was used, satisfied $L(k)-L(k-1) < 10^{-5}(L(k-1)-L(2))$. At the end of learning, the log likelihoods on the training and test set were computed for all models using the exact algorithm. Also included in the comparison was the log likelihood of the training and test sets under the true model that generated the data. The test set log likelihood for N observation sequences is defined as $\sum_{n=1}^{N} \log P(Y_1^{(n)}, \ldots, Y_T^{(n)}|\phi)$, where ϕ is obtained by maximizing the log likelihood over a training set that is disjoint from the test set. This provides a measure of how well the model generalizes to a novel observation sequence from the same distribution as the training data.

Results averaged over 15 runs for each algorithm on each of the four problem sizes (a total of 300 runs) are presented in Table 1. Even for the smallest problem size ($M = 3$ and $K = 2$), the standard HMM with K^M states suffers from overfitting: the test set log likelihood is significantly worse than the training set log likelihood. As expected, this overfitting problem becomes worse as the size of the state space increases; it is particularly serious for $M = 5$ and $K = 3$.

For the factorial HMMs, the log likelihoods for each of the three approximate EM algorithms were compared to the exact algorithm. Gibbs sampling appeared to have the poorest performance: for each of the three smaller size problems its log likelihood was significantly worse than that of the exact algorithm on both the training sets and test sets ($p < 0.05$). This may be due to insufficient sampling. However, we will soon see that running the Gibbs sampler for more than 10 samples, while potentially improving performance, makes it substantially slower than the variational methods. Surprisingly, the Gibbs sampler appears to do quite well on the largest size problem, although the differences to the other methods are not statistically significant.

The performance of the completely factorized variational approximation was not statistically significantly different from that of the exact algorithm on either the training set or the test set for any of the problem sizes. The performance of the structured variational approximation was not statistically different from that of the exact method on three of the four problem sizes, and appeared to be *better* on one of the problem sizes ($M = 5,\ K = 2;\ p < 0.05$). Although this result may be a fluke arising from random variability, there is another more interesting (and speculative) explanation. The exact EM algorithm implements unconstrained maximization of F, as defined in section 3.6, while the variational methods maximize F subject to a constrained distribution. These constraints could presumably act as regularizers, reducing overfitting.

There was a large amount of variability in the final log likelihoods for the models learned by all the algorithms. We subtracted the log likelihood of the true generative model from that of each trained model to eliminate the main effect of the randomly sampled generative model and to reduce the variability due to training and test sets. One important remaining source of variance was the random seed used in

Table 1. Comparison of the factorial HMM on four problems of varying size. The negative log likelihood for the training and test set, plus or minus one standard deviation, is shown for each problem size and algorithm, measured in bits per observation (log likelihood in bits divided by NT) relative to the log likelihood under the true generative model for that data set.[7] *True* is the true generative model (the log likelihood per symbol is defined to be zero for this model by our measure); *HMM* is the hidden Markov model with K^M states; *Exact* is the factorial HMM trained using an exact E step; *Gibbs* is the factorial HMM trained using Gibbs sampling; *CFVA* is the factorial HMM trained using the completely factorized variational approximation; *SVA* is the factorial HMM trained using the structured variational approximation.

M	K	Algorithm	Training Set	Test Set
3	2	*True*	0.00 ± 0.39	0.00 ± 0.39
		HMM	1.19 ± 0.67	2.29 ± 1.02
		Exact	0.88 ± 0.80	1.05 ± 0.72
		Gibbs	1.67 ± 1.23	1.78 ± 1.22
		CFVA	1.06 ± 1.20	1.20 ± 1.11
		SVA	0.91 ± 1.02	1.04 ± 1.01
3	3	*True*	0.00 ± 0.19	0.00 ± 0.20
		HMM	0.76 ± 0.67	9.81 ± 2.55
		Exact	1.02 ± 1.04	1.26 ± 0.99
		Gibbs	2.21 ± 0.91	2.50 ± 0.87
		CFVA	1.24 ± 1.50	1.50 ± 1.53
		SVA	0.64 ± 0.88	0.90 ± 0.84
5	2	*True*	0.00 ± 0.60	0.00 ± 0.57
		HMM	0.83 ± 0.82	11.57 ± 3.71
		Exact	2.29 ± 1.19	2.51 ± 1.21
		Gibbs	3.25 ± 1.17	3.35 ± 1.14
		CFVA	1.73 ± 1.34	2.07 ± 1.74
		SVA	1.34 ± 1.07	1.53 ± 1.05
5	3	*True*	0.00 ± 0.30	0.00 ± 0.29
		HMM	-4.80 ± 0.52	175.35 ± 84.74
		Exact	4.23 ± 2.28	4.49 ± 2.24
		Gibbs	3.63 ± 1.13	3.95 ± 1.14
		CFVA	4.85 ± 0.68	5.14 ± 0.69
		SVA	3.99 ± 1.57	4.30 ± 1.62

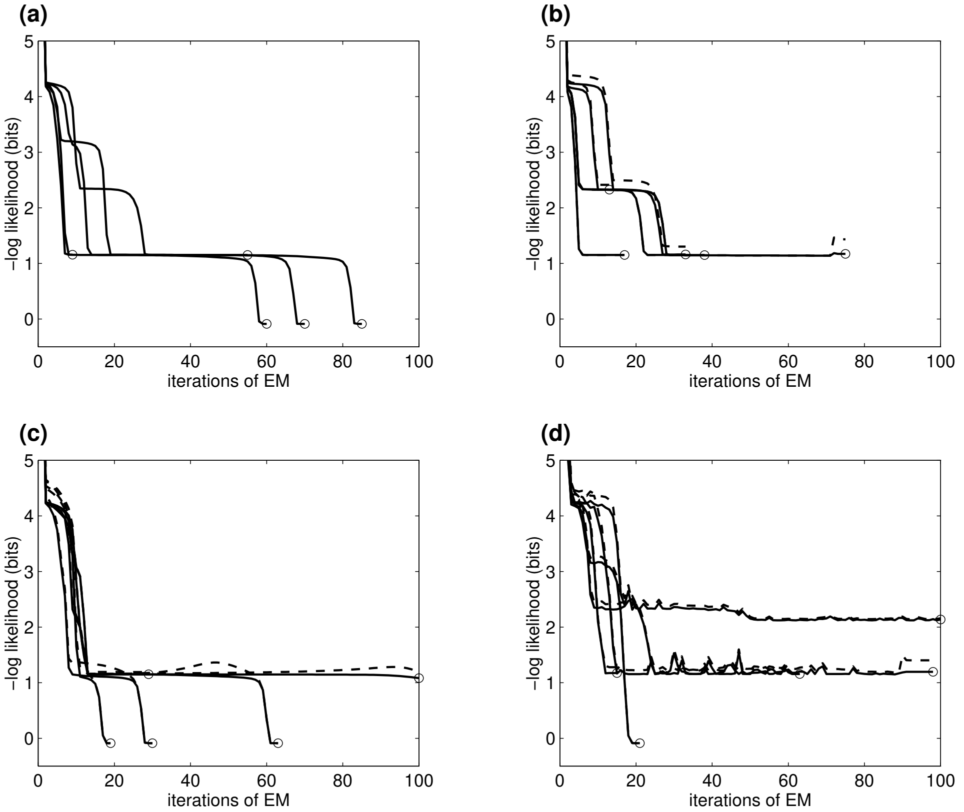

Figure 3. Learning curves for five runs of each of the four learning algorithms for factorial HMMs: (a) exact; (b) completely factorized variational approximation; (c) structured variational approximation; and (d) Gibbs sampling. A single training set sampled from the $M = 3$, $K = 2$ problem size was used for all these runs. The solid lines show the negative log likelihood per observation (in bits) relative to the true model that generated the data, calculated using the exact algorithm. The circles denote the point at which the convergence criterion was met and the run ended. For the three approximate algorithms, the dashed lines show an approximate negative log likelihood.[8]

each training run, which determined the initial parameters and the samples used in the Gibbs algorithm. All algorithms appeared to be very sensitive to this random seed, suggesting that different runs on each training set found different local maxima or plateaus of the likelihood (Figure 3). Some of this variability could be eliminated by explicitly adding a regularization term, which can be viewed as a prior on the parameters in maximum a posteriori parameter estimation. Alternatively, Bayesian (or ensemble) methods could be used to average out this variability by integrating over the parameter space.

The timing comparisons confirm the fact that both the standard HMM and the exact E step factorial HMM are extremely slow for models with large state spaces (Fig-

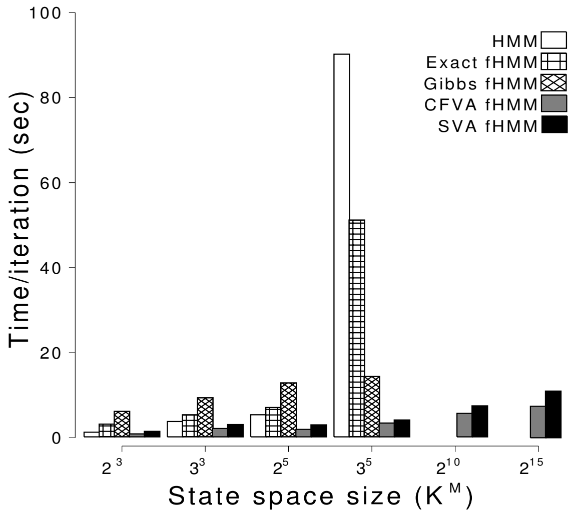

Figure 4. Time per iteration of EM on a Silicon Graphics R4400 processor running Matlab.

ure 4). Gibbs sampling was slower than the variational methods even when limited to ten samples of each hidden variable per iteration of EM. Since one pass of the variational fixed point equations has the same time complexity as one pass of Gibbs sampling, and since the variational fixed point equations were found to converge very quickly, these experiments suggest that Gibbs sampling is not as competitive time-wise as the variational methods. The time per iteration for the variational methods scaled well to large state spaces.

4.2. Experiment 2: Bach chorales

Musical pieces naturally exhibit complex structure at many different time scales. Furthermore, one can imagine that to represent the "state" of the musical piece at any given time it would be necessary to specify a conjunction of many different features. For these reasons, we chose to test whether a factorial HMM would provide an advantage over a regular HMM in modeling a collection of musical pieces.

The data set consisted of discrete event sequences encoding the melody lines of J. S. Bach's Chorales, obtained from the UCI Repository for Machine Learning Databases (Merz & Murphy, 1996) and originally discussed in Conklin and Witten (1995). Each event in the sequence was represented by six attributes, described in Table 2. Sixty-six chorales, with 40 or more events each, were divided into a training set (30 chorales) and a test set (36 chorales). Using the first set, hidden Markov models with state space ranging from 2 to 100 states were trained until convergence (30 ± 12 steps of EM). Factorial HMMs of varying sizes (K ranging from 2 to 6; M ranging from 2 to 9) were also trained on the same data. To

Table 2. Attributes in the Bach chorale data set. The key signature and time signature attributes were constant over the duration of the chorale. All attributes were treated as real numbers and modeled as linear-Gaussian observations (4a).

Attribute	Description	Representation
pitch	pitch of the event	int $[0, 127]$
keysig	key signature	int $[-7, 7]$
timesig	time signature	(1/16 notes)
fermata	event under fermata?	binary
st	start time of event	int (1/16 notes)
dur	duration of event	int (1/16 notes)

approximate the E step for factorial HMMs we used the structured variational approximation. This choice was motivated by three considerations. First, for the size of state space we wished to explore, the exact algorithms were prohibitively slow. Second, the Gibbs sampling algorithm did not appear to present any advantages in speed or performance and required some heuristic method for determining the number of samples. Third, theoretical arguments suggest that the structured approximation should in general be superior to the completely factorized variational approximation, since more of the dependencies of the original model are preserved.

The test set log likelihoods for the HMMs, shown in Figure 5 (a), exhibited the typical U-shaped curve demonstrating a trade-off between bias and variance (Geman, Bienenstock, & Doursat, 1992). HMMs with fewer than 10 states did not predict well, while HMMs with more than 40 states overfit the training data and therefore provided a poor model of the test data. Out of the 75 runs, the highest test set log likelihood per observation was -9.0 bits, obtained by an HMM with 30 hidden states.[9]

The factorial HMM provides a more satisfactory model of the chorales from three points of view. First, the time complexity is such that it is possible to consider models with significantly larger state spaces; in particular, we fit models with up to 1000 states. Second, given the componential parametrization of the factorial HMM, these large state spaces do not require excessively large numbers of parameters relative to the number of data points. In particular, we saw no evidence of overfitting even for the largest factorial HMM as seen in Figures 5 (c) & (d). Finally, this approach resulted in significantly better predictors; the test set likelihood for the best factorial HMM was an order of magnitude larger than the test set likelihood for the best HMM, as Figure 5 (d) reveals.

While the factorial HMM is clearly a better predictor than a single HMM, it should be acknowledged that neither approach produces models that are easily interpretable from a musicological point of view. The situation is reminiscent of that in speech recognition, where HMMs have proved their value as predictive models of the speech signal without necessarily being viewed as causal generative models of speech. A factorial HMM is clearly an impoverished representation of

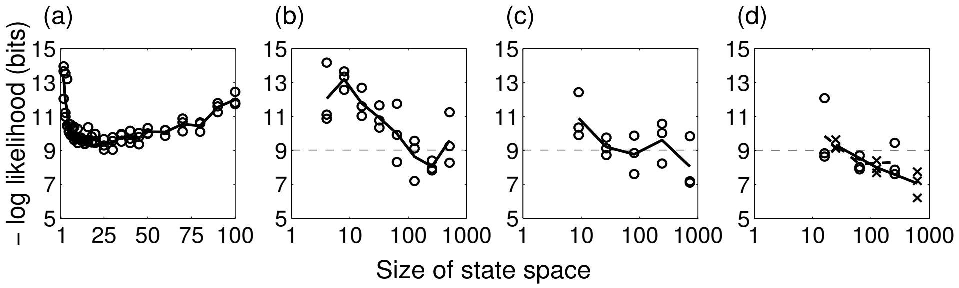

Figure 5. Test set log likelihood per event of the Bach chorale data set as a function of number of states for (a) HMMs, and factorial HMMs with (b) $K = 2$, (c) $K = 3$, and (d) $K = 4$ (o's; heavy dashed line) and $K = 5$ ($\times$'s; solid line). Each symbol represents a single run; the lines indicate the mean performances. The thin dashed line in (b)–(d) indicates the log likelihood per observation of the best run in (a). The factorial HMMs were trained using the structured approximation. For both methods the true likelihood was computed using the exact algorithm.

musical structure, but its promising performance as a predictor provides hope that it could serve as a step on the way toward increasingly structured statistical models for music and other complex multivariate time series.

5. Generalizations of the model

In this section, we describe four variations and generalizations of the factorial HMM.

5.1. Discrete observables

The probabilistic model presented in this paper has assumed real-valued Gaussian observations. One of the advantages arising from this assumption is that the conditional density of a D-dimensional observation, $P(Y_t|S_t^{(1)}, \ldots, S_t^{(M)})$, can be compactly specified through M mean matrices of dimension $D \times K$, and one $D \times D$ covariance matrix. Furthermore, the M step for such a model reduces to a set of weighted least squares equations.

The model can be generalized to handle discrete observations in several ways. Consider a single D-valued discrete observation Y_t. In analogy to traditional HMMs, the output probabilities could be modeled using a matrix. However, in the case of a factorial HMM, this matrix would have $D \times K^M$ entries for each combination of the state variables and observation. Thus the compactness of the representation would be entirely lost. Standard methods from graphical models suggest approximating such large matrices with "noisy-OR" (Pearl, 1988) or "sigmoid" (Neal, 1992) models of interaction. For example, in the softmax model, which generalizes the sigmoid model to $D > 2$, $P(Y_t|S_t^{(1)}, \ldots, S_t^{(M)})$ is multinomial with mean proportional to

$\exp\left\{\sum_m W^{(m)} S_t^{(m)}\right\}$. Like the Gaussian model, this specification is again compact, using M matrices of size $D \times K$. (As in the linear-Gaussian model, the $W^{(m)}$ are overparametrized since they can each model the overall mean of Y_t, as shown in Appendix A.) While the nonlinearity induced by the softmax function makes both the E step and M step of the algorithm more difficult, iterative numerical methods can be used in the M step whereas Gibbs sampling and variational methods can again be used in the E step (see Neal, 1992; Hinton et al., 1995; and Saul et al., 1996, for discussions of different approaches to learning in sigmoid networks).

5.2. Introducing couplings

The architecture for factorial HMMs presented in Section 2 assumes that the underlying Markov chains interact only through the observations. This constraint can be relaxed by introducing couplings between the hidden state variables (cf. Saul & Jordan, 1997). For example, if $S_t^{(m)}$ depends on $S_{t-1}^{(m)}$ and $S_{t-1}^{(m-1)}$, equation (3) is replaced by the following factorization

$$P(S_t|S_{t-1}) = P(S_t^{(1)}|S_{t-1}^{(1)}) \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}, S_{t-1}^{(m-1)}). \tag{13}$$

Similar exact, variational, and Gibbs sampling procedures can be defined for this architecture. However, note that these couplings must be introduced with caution, as they may result in an exponential growth in parameters. For example, the above factorization requires transition matrices of size $K^2 \times K$. Rather than specifying these higher-order couplings through probability transition matrices, one can introduce second-order interaction terms in the energy (log probability) function. Such terms effectively couple the chains without the number of parameters incurred by a full probability transition matrix.[10] In the graphical model formalism these correspond to symmetric undirected links, making the model a chain graph. While the Jensen, Lauritzen and Olesen (1990) algorithm can still be used to propagate information exactly in chain graphs, such undirected links cause the normalization constant of the probability distribution—the *partition function*—to depend on the coupling parameters. As in Boltzmann machines (Hinton & Sejnowski, 1986), both a clamped and an unclamped phase are therefore required for learning, where the goal of the unclamped phase is to compute the derivative of the partition function with respect to the parameters (Neal, 1992).

5.3. Conditioning on inputs

Like the hidden Markov model, the factorial HMM provides a model of the unconditional density of the observation sequences. In certain problem domains, some of the observations can be better thought of as inputs or explanatory variables, and the others as outputs or response variables. The goal, in these cases, is to model the conditional density of the output sequence given the input sequence. In machine learning terminology, unconditional density estimation is unsupervised while conditional density estimation is supervised.

Several algorithms for learning in hidden Markov models that are conditioned on inputs have been recently presented in the literature (Cacciatore & Nowlan, 1994; Bengio & Frasconi, 1995; Meila & Jordan, 1996). Given a sequence of input vectors $\{X_t\}$, the probabilistic model for an input-conditioned factorial HMM is

$$\begin{aligned} P(\{S_t, Y_t\}|\{X_t\}) &= \prod_{m=1}^{M} P(S_1^{(m)}|X_1)P(Y_1|S_1^{(m)}, X_1) \\ &\quad \times \prod_{t=2}^{T}\prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}, X_t)P(Y_t|S_t^{(m)}, X_t). \end{aligned} \tag{14}$$

The model depends on the specification of $P(Y_t|S_t^{(m)}, X_t)$ and $P(S_t^{(m)}|S_{t-1}^{(m)}, X_t)$, which are conditioned both on a discrete state variable and on a (possibly continuous) input vector. The approach used in Bengio and Frasconi's Input Output HMMs (IOHMMs) suggests modeling $P(S_t^{(m)}|S_{t-1}^{(m)}, X_t)$ as K separate neural networks, one for each setting of $S_{t-1}^{(m)}$. This decomposition ensures that a valid probability transition matrix is defined at each point in input space if a sum-to-one constraint (e.g., softmax nonlinearity) is used in the output of these networks.

Using the decomposition of each conditional probability into K networks, inference in input-conditioned factorial HMMs is a straightforward generalization of the algorithms we have presented for factorial HMMs. The exact forward–backward algorithm in Appendix B can be adapted by using the appropriate conditional probabilities. Similarly, the Gibbs sampling procedure is no more complex when conditioned on inputs. Finally, the completely factorized and structured approximations can also be generalized readily if the approximating distribution has a dependence on the input similar to the model's. If the probability transition structure $P(S_t^{(m)}|S_{t-1}^{(m)}, X_t)$ is not decomposed as above, but has a complex dependence on the previous state variable and input, inference may become considerably more complex.

Depending on the form of the input conditioning, the Maximization step of learning may also change considerably. In general, if the output and transition probabilities are modeled as neural networks, the M step can no longer be solved exactly and a gradient-based generalized EM algorithm must be used. For log-linear models, the M step can be solved using an inner loop of iteratively reweighted least-squares (McCullagh & Nelder, 1989).

5.4. Hidden Markov decision trees

An interesting generalization of factorial HMMs results if one conditions on an input X_t and orders the M state variables such that $S_t^{(m)}$ depends on $S_t^{(n)}$ for

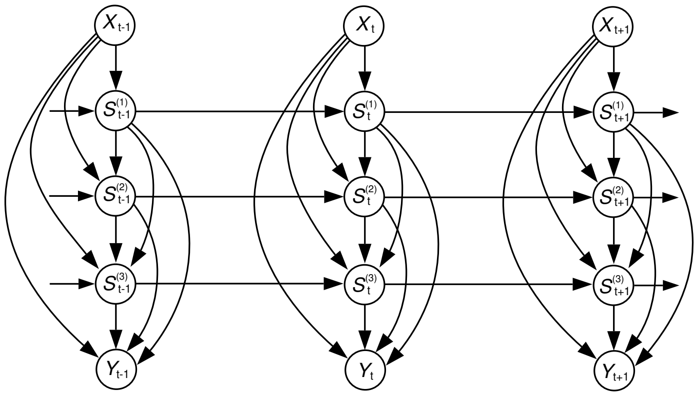

Figure 6. The hidden Markov decision tree.

$1 \leq n < m$ (Figure 6). The resulting architecture can be seen as a probabilistic decision tree with Markovian dynamics linking the decision variables. Consider how this probabilistic model would generate data at the first time step, $t = 1$. Given input X_1, the top node $S_1^{(1)}$ can take on K values. This stochastically partitions X space into K decision regions. The next node down the hierarchy, $S_1^{(2)}$, subdivides each of these regions into K subregions, and so on. The output Y_1 is generated from the input X_1 and the K-way decisions at each of the M hidden nodes. At the next time step, a similar procedure is used to generate data from the model, except that now each decision in the tree is dependent on the decision taken at that node in the previous time step. Thus, the "hierarchical mixture of experts" architecture (Jordan & Jacobs, 1994) is generalized to include Markovian dynamics for the decisions. Hidden Markov decision trees provide a useful starting point for modeling time series with both temporal and spatial structure at multiple resolutions. We explore this generalization of factorial HMMs in Jordan, Ghahramani, and Saul (1997).

6. Conclusion

In this paper we have examined the problem of learning for a class of generalized hidden Markov models with distributed state representations. This generalization provides both a richer modeling tool and a method for incorporating prior structural information about the state variables underlying the dynamics of the system generating the data. Although exact inference in this class of models is generally intractable, we provided a structured variational approximation that can be computed tractably. This approximation forms the basis of the Expectation step in an EM algorithm for learning the parameters of the model. Empirical comparisons to several other approximations and to the exact algorithm show that this approximation is both efficient to compute and accurate. Finally, we have shown that the factorial HMM representation provides an advantage over traditional HMMs in predictive modeling of the complex temporal patterns in Bach's chorales.

Appendix A

The M step

The M step equations for each parameter are obtained by setting the derivatives of $\mathcal{Q}$ with respect to that parameters to zero. We start by expanding $\mathcal{Q}$ using equations (1)–(4b):

$$\begin{aligned}\mathcal{Q} \;=\; & -\frac{1}{2}\sum_{t=1}^{T}\Bigg[Y_t' C^{-1} Y_t - 2\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} \langle S_t^{(m)}\rangle \\ & \qquad\qquad + \sum_{m=1}^{M}\sum_{n=1}^{M} tr\left\{W^{(m)'} C^{-1} W^{(n)} \langle S_t^{(n)} S_t^{(m)'}\rangle\right\}\Bigg] \\ & + \sum_{m=1}^{M}\langle S_1^{(m)'}\rangle \log \pi^{(m)} + \sum_{t=2}^{T}\sum_{m=1}^{M} tr\left\{(\log P^{(m)})\langle S_{t-1}^{(m)} S_t^{(m)'}\rangle\right\} - \log Z, \end{aligned} \tag{A.1}$$

where tr is the trace operator for square matrices and Z is a normalization term independent of the states and observations ensuring that the probabilities sum to one.

Setting the derivatives of $\mathcal{Q}$ with respect to the output weights to zero, we obtain a linear system of equations for the $W^{(m)}$:

$$\frac{\partial \mathcal{Q}}{\partial W^{(m)}} = \sum_{t=1}^{T}\left[\sum_{n=1}^{M} W^{(n)} \langle S_t^{(n)} S_t^{(m)'}\rangle - Y_t \langle S_t^{(m)'}\rangle\right] = 0. \tag{A.2}$$

Assuming Y_t is a $D\times 1$ vector, let S_t be the $MK\times 1$ vector obtained by concatenating the $S^{(m)}$ vectors, and W be the $D \times MK$ matrix obtained by concatenating the $W^{(m)}$ matrices (of size $D \times K$). Then solving (A.2) results in

$$W^{\text{new}} = \left(\sum_{t=1}^{T} Y_t \langle S_t'\rangle\right)\left(\sum_{t=1}^{T}\langle S_t S_t'\rangle\right)^{\dagger}, \tag{A.3}$$

where $\dagger$ is the Moore-Penrose pseudo-inverse. Note that the model is overparameterized since the $D\times 1$ means of each of the $W^{(m)}$ matrices add up to a single mean. Using the pseudo-inverse removes the need to explicitly subtract this overall mean from each $W^{(m)}$ and estimate it separately as another parameter.

To estimate the priors, we solve $\partial\mathcal{Q}/\partial\pi^{(m)} = 0$ subject to the constraint that they sum to one, obtaining

$$\pi^{(m)\,\text{new}} = \langle S_1^{(m)}\rangle. \tag{A.4}$$

Similarly, to estimate the transition matrices we solve $\partial \mathcal{Q}/\partial P^{(m)} = 0$, subject to the constraint that the columns of $P^{(m)}$ sum to one. For element (i,j) in $P^{(m)}$,

$$P_{i,j}^{(m)\,\text{new}} = \frac{\sum_{t=2}^{T} \langle S_{t,i}^{(m)} S_{t-1,j}^{(m)} \rangle}{\sum_{t=2}^{T} \langle S_{t-1,j}^{(m)} \rangle}. \tag{A.5}$$

Finally, the re-estimation equations for the covariance matrix can be derived by taking derivatives with respect to C^{-1}

$$\frac{\partial \mathcal{Q}}{\partial C^{-1}} = \frac{T}{2}C + \sum_{t=1}^{T} \left[\sum_{m=1}^{M} Y_t \langle S_t^{(m)'} \rangle W^{(m)'} - \frac{1}{2} Y_t Y_t' - \frac{1}{2} \sum_{m,n=1}^{M} W^{(n)} \langle S_t^{(n)} S_t^{(m)'} \rangle W^{(m)'} \right]. \tag{A.6}$$

The first term arises from the normalization for the Gaussian density function: Z is proportional to $|C|^{T/2}$ and $\partial |C| / \partial C^{-1} = C$. Substituting (A.2) and re-organizing we get

$$C^{\text{new}} = \frac{1}{T} \sum_{t=1}^{T} Y_t Y_t' - \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle \, Y_t'. \tag{A.7}$$

For $M = 1$, these equations reduce to the Baum-Welch re-estimation equations for HMMs with Gaussian observables. The above M step has been presented for the case of a single observation sequence. The extension to multiple sequences is straightforward.

Appendix B

Exact forward–backward algorithm

Here we specify an exact forward–backward recursion for computing the posterior probabilities of the hidden states in a factorial HMM. It differs from a straightforward application of the forward–backward algorithm on the equivalent K^M state HMM, in that it does not depend on a $K^M \times K^M$ transition matrix. Rather, it makes use of the independence of the underlying Markov chains to sum over M transition matrices of size $K \times K$.

Using the notation $\{Y_\tau\}_t^r$ to mean the observation sequence $Y_t, \ldots, Y_r$, we define

$$\begin{aligned}
\alpha_t &= P(S_t^{(1)}, S_t^{(2)}, \ldots S_t^{(M)}, \{Y_\tau\}_1^t | \phi) \\
\alpha_t^{(0)} &= P(S_t^{(1)}, S_t^{(2)}, \ldots S_t^{(M)}, \{Y_\tau\}_1^{t-1} | \phi) \\
\alpha_t^{(1)} &= P(S_{t-1}^{(1)}, S_t^{(2)}, \ldots S_t^{(M)}, \{Y_\tau\}_1^{t-1} | \phi) \\
&\vdots \\
\alpha_t^{(M)} &= P(S_{t-1}^{(1)}, \ldots S_{t-1}^{(M)}, \{Y_\tau\}_1^{t-1} | \phi) = \alpha_{t-1} \quad .
\end{aligned}$$

Then we obtain the forward recursions

$$\alpha_t = P(Y_t | S_t^{(1)}, \ldots, S_t^{(M)}, \phi) \alpha_t^{(0)} \tag{B.1}$$

and

$$\alpha_t^{(m-1)} = \sum_{S_{t-1}^{(m)}} P(S_t^{(m)} | S_{t-1}^{(m)}) \alpha_t^{(m)} \quad . \tag{B.2}$$

At the end of the forward recursions, the likelihood of the observation sequence is the sum of the K^M elements in α_T.

Similarly, to obtain the backward recursions we define

$$\begin{aligned}
\beta_t &= P(\{Y_\tau\}_{t+1}^T | S_t^{(1)}, \ldots S_t^{(M)}, \phi) \\
\beta_{t-1}^{(M)} &= P(\{Y_\tau\}_t^T | S_t^{(1)}, \ldots S_t^{(M)}, \phi) \\
&\vdots \\
\beta_{t-1}^{(1)} &= P(\{Y_\tau\}_t^T | S_t^{(1)}, S_{t-1}^{(2)} \ldots S_{t-1}^{(M)}, \phi) \\
\beta_{t-1}^{(0)} &= P(\{Y_\tau\}_t^T | S_{t-1}^{(1)}, S_{t-1}^{(2)} \ldots S_{t-1}^{(M)}, \phi) = \beta_{t-1},
\end{aligned}$$

from which we obtain

$$\beta_{t-1}^{(M)} = P(Y_t | S_t^{(1)}, \ldots, S_t^{(M)}, \phi) \beta_t \tag{B.3}$$

$$\beta_{t-1}^{(m-1)} = \sum_{S_t^{(m)}} P(S_t^{(m)} | S_{t-1}^{(m)}) \beta_{t-1}^{(m)} \quad . \tag{B.4}$$

The posterior probability of the state at time t is obtained by multiplying α_t and β_t and normalizing:

$$\gamma_t = P(S_t | \{Y_\tau\}_1^T, \phi) = \frac{\alpha_t \beta_t}{\sum_{S_t} \alpha_t \beta_t} \quad . \tag{B.5}$$

This algorithm can be shown to be equivalent to the Jensen, Lauritzen and Olesen algorithm for probability propagation in graphical models. The probabilities are defined over collections of state variables corresponding to the cliques in the equivalent junction tree. Information is passed forwards and backwards by summing over the sets separating each neighboring clique in the tree. This results in forward–backward-type recursions of order $O(TMK^{M+1})$.

Using the α_t, β_t, and γ_t quantities, the statistics required for the E step are

$$\langle S_t^{(m)} \rangle = \sum_{S_t^{(n)} (n \neq m)} \gamma_t \tag{B.6}$$

$$\langle S_t^{(m)} S_t^{(n)\prime} \rangle = \sum_{S_t^{(r)} (r \neq m \wedge r \neq n)} \gamma_t \tag{B.7}$$

$$\langle S_{t-1}^{(m)} S_t^{(m)'} \rangle = \frac{\displaystyle\sum_{S_{t-1}^{(n)}, S_t^{(r)} (n \neq m \wedge r \neq m)} \alpha_{t-1} P(S_t|S_{t-1}) P(Y_t|S_t) \beta_t}{\displaystyle\sum_{S_{t-1}, S_t} \alpha_{t-1} P(S_t|S_{t-1}) P(Y_t|S_t) \beta_t} \quad . \tag{B.8}$$

Appendix C

Completely factorized variational approximation

Using the definition of the probabilistic model given by equations (1)–(4b), the posterior probability of the states given an observation sequence can be written as

$$P(\{S_t\}|\{Y_t\}, \phi) = \frac{1}{Z} \exp\{-H(\{S_t, Y_t\})\} \quad , \tag{C.1}$$

where Z is a normalization constant ensuring that the probabilities sum to one and

$$\begin{aligned} H(\{S_t, Y_t\}) = {} & \frac{1}{2} \sum_{t=1}^{T} \left(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \right)' C^{-1} \left(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \right) \\ & - \sum_{m=1}^{M} S_1^{(m)'} \log \pi^{(m)} - \sum_{t=2}^{T} \sum_{m=1}^{M} S_t^{(m)'} (\log P^{(m)}) S_{t-1}^{(m)} \quad . \end{aligned} \tag{C.2}$$

Similarly, the probability distribution given by the variational approximation (7)–(8) can be written as

$$Q(\{S_t\}|\theta) = \frac{1}{Z_Q} \exp\{-H_Q(\{S_t\})\} \quad , \tag{C.3}$$

where

$$H_Q(\{S_t\}) = -\sum_{t=1}^{T} \sum_{m=1}^{M} S_t^{(m)'} \log \theta_t^{(m)} \quad . \tag{C.4}$$

Using this notation, and denoting expectation with respect to the variational distribution using angular brackets $\langle \cdot \rangle$, the KL divergence is

$$\mathrm{KL}(Q \| P) = \langle H \rangle - \langle H_Q \rangle - \log Z_Q + \log Z. \tag{C.5}$$

Three facts can be verified from the definition of the variational approximation:

$$\langle S_t^{(m)} \rangle = \theta_t^{(m)} \tag{C.6}$$

$$\langle S_{t-1}^{(m)} S_t^{(m)'} \rangle = \theta_{t-1}^{(m)} \theta_t^{(m)\prime} \tag{C.7}$$

$$\langle S_t^{(m)} S_t^{(n)'} \rangle = \begin{cases} \theta_t^{(m)} \theta_t^{(n)'} & \text{if } m \neq n \\ diag\{\theta_t^{(m)}\} & \text{if } m = n \end{cases} \tag{C.8}$$

where *diag* is an operator that takes a vector and returns a square matrix with the elements of the vector along its diagonal, and zeros everywhere else. The KL divergence can therefore be expanded to

$$\begin{aligned}
\mathrm{KL} = & \sum_{t=1}^{T}\sum_{m=1}^{M} \theta_t^{(m)'} \log \theta_t^{(m)} + \frac{1}{2}\sum_{t=1}^{T}\Bigg[Y_t' C^{-1} Y_t - 2\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} \theta_t^{(m)} \\
& + \sum_{m=1}^{M}\sum_{n\neq m}^{M} tr\{W^{(m)'} C^{-1} W^{(n)} \theta_t^{(n)} \theta_t^{(m)'}\} + \sum_{m=1}^{M} tr\Big\{W^{(m)'} C^{-1} W^{(m)} diag\{\theta_t^{(m)}\}\Big\}\Bigg] \\
& + \sum_{m=1}^{M} \theta_1^{(m)'} \log \pi^{(m)} + \sum_{t=2}^{T}\sum_{m=1}^{M} tr\{\theta_{t-1}^{(m)} \theta_t^{(m)'} \log P^{(m)}\} - \log Z_Q + \log Z. \qquad \text{(C.9)}
\end{aligned}$$

Taking derivatives with respect to $\theta_t^{(m)}$, we obtain

$$\begin{aligned}
\frac{\partial \mathrm{KL}}{\partial \theta_t^{(m)}} = & \log \theta_t^{(m)} - W^{(m)'} C^{-1} Y_t + \sum_{n\neq m}^{M} W^{(m)'} C^{-1} W^{(n)} \theta_t^{(n)} + \frac{1}{2}\Delta^{(m)} \\
& -(\log P^{(m)})\ \theta_{t-1}^{(m)} - (\log P^{(m)})'\ \theta_{t+1}^{(m)} + c \quad , \qquad \text{(C.10)}
\end{aligned}$$

where $\Delta^{(m)}$ is the vector of diagonal elements of $W^{(m)'} C^{-1} W^{(m)}$, and c is a term arising from $\log Z_Q$, ensuring that the $\theta_t^{(m)}$ sum to one. Setting this derivative equal to 0 and solving for $\theta_t^{(m)}$ gives equation (9a).

Appendix D
Structured approximation

For the structured approximation, H_Q is defined as

$$\begin{aligned}
H_Q(\{S_t\}) = & -\sum_{m=1}^{M} S_1^{(m)'} \log \pi^{(m)} - \sum_{t=2}^{T}\sum_{m=1}^{M} S_t^{(m)'} (\log P^{(m)}) S_{t-1}^{(m)} \\
& -\sum_{t=1}^{T}\sum_{m=1}^{M} S_t^{(m)'} \log h_t^{(m)}. \qquad \text{(D.1)}
\end{aligned}$$

Using (C.2), we write the KL divergence as

$$\begin{aligned}
\mathrm{KL} = & \sum_{t=1}^{T}\sum_{m=1}^{M} \langle S_t^{(m)}\rangle \log h_t^{(m)} + \frac{1}{2}\sum_{t=1}^{T}\Bigg[Y_t' C^{-1} Y_t - 2\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} \langle S_t^{(m)}\rangle \\
& + \sum_{m=1}^{M}\sum_{n\neq m}^{M} tr\Big\{W^{(m)'} C^{-1} W^{(n)} \langle S_t^{(n)}\rangle \langle S_t^{(m)'}\rangle\Big\} \\
& + \sum_{m=1}^{M} tr\Big\{W^{(m)'} C^{-1} W^{(m)} diag\Big\{\langle S_t^{(m)}\rangle\Big\}\Big\}\Bigg] - \log Z_Q + \log Z. \qquad \text{(D.2)}
\end{aligned}$$

Since KL is independent of $\pi^{(m)}$ and $P^{(m)}$, the first thing to note is that these parameters of the structured approximation remain equal to the equivalent parameters of the true system. Now, taking derivatives with respect to $\log h_\tau^{(n)}$, we get

$$\begin{aligned}\frac{\partial \mathrm{KL}}{\partial \log h_\tau^{(n)}} &= \langle S_\tau^{(n)}\rangle + \sum_{t=1}^{T}\sum_{m=1}^{M}\Bigg[\log h_t^{(m)} - W^{(m)\prime}C^{-1}Y_t + \sum_{\ell\neq m}^{M} W^{(m)\prime}C^{-1}W^{(\ell)}\langle S_t^{(\ell)}\rangle \\ &\qquad + \frac{1}{2}\Delta^{(m)}\Bigg]\frac{\partial\langle S_t^{(m)}\rangle}{\partial \log h_\tau^{(n)}} - \langle S_\tau^{(n)}\rangle. \qquad \text{(D.3)}\end{aligned}$$

The last term, which we obtained by making use of the fact that

$$\frac{\partial \log Z_Q}{\partial \log h_\tau^{(n)}} = \langle S_\tau^{(n)}\rangle, \qquad \text{(D.4)}$$

cancels out the first term. Setting the terms inside the brackets in (D.3) equal to zero yields equation (12a).

Acknowledgments

We thank Lawrence Saul for helpful discussions and Geoffrey Hinton for support. This project was supported in part by a grant from the McDonnell-Pew Foundation, by a grant from ATR Human Information Processing Research Laboratories, by a gift from Siemens Corporation, and by grant N00014-94-1-0777 from the Office of Naval Research. Zoubin Ghahramani was supported by a grant from the Ontario Information Technology Research Centre.

Notes

1. For related work on inference in distributed state HMMs, see Dean and Kanazawa (1989).
2. In speech, neural networks are generally used to model $P(S_t|Y_t)$; this probability is converted to the observation probabilities needed in the HMM via Bayes rule.
3. If the columns of $W^{(m)}$ and $W^{(n)}$ are orthogonal for every pair of state variables, m and n, and C is a diagonal covariance matrix, then the state variables will no longer be dependent given the observation. In this case there is no “explaining away”: each state variable is modeling the variability in the observation along a different subspace.
4. A more Bayesian treatment of the learning problem, in which the parameters are also considered hidden random variables, can be handled by Gibbs sampling by replacing the “M step” with sampling from the conditional distribution of the parameters given the other hidden variables (for example, see Tanner and Wong, 1987).
5. The first term is replaced by $\log \pi^{(m)}$ for $t = 1$ the second term does not appear for $t = T$.
6. All samples were used for learning; that is, no samples were discarded at the beginning of the run. Although ten samples is too few to even approach convergence, it provides a run-time roughly comparable to the variational methods. The goal was to see whether this “impatient” Gibbs sampler would be able to compete with the other approximate methods.

7. Lower values suggest a better probabilistic model: a value of one, for example, means that it would take one bit more than the true generative model to code each observation vector. Standard deviations reflect the variation due to training set, test set, and the random seed of the algorithm. Standard errors on the mean are a factor of 3.8 smaller.
8. For the variational methods these dashed lines are equal to minus the lower bound on the log likelihood, except for a normalization term which is intractable to compute and can vary during learning, resulting in the apparent occasional increases in the bound.
9. Since the attributes were modeled as real numbers, the log likelihoods are only a measure of relative coding cost. Comparisons between these likelihoods are meaningful, whereas to obtain the absolute cost of coding a sequence, it is necessary to specify a discretization level.
10. This is analogous to the fully-connected Boltzmann machine with N units (Hinton & Sejnowski, 1986), in which every binary unit is coupled to every other unit using $O(N^2)$ parameters, rather than the $O(2^N)$ parameters required to specify the complete probability table.

References

Baum, L., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, *41*, 164–171.

Bengio, Y., & Frasconi, P. (1995). An input–output HMM architecture. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems 7*, pp. 427–434. Cambridge, MA: MIT Press.

Cacciatore, T. W., & Nowlan, S. J. (1994). Mixtures of controllers for jump linear and non-linear plants. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems 6*, pp. 719–726. San Francisco, CA: Morgan Kaufmann.

Conklin, D., & Witten, I. H. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, *24*, 51–73.

Cover, T., & Thomas, J. (1991). *Elements of information theory.* New York: John Wiley.

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, *2*, 25–36.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*, 142–150.

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

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*, 1–58.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

Ghahramani, Z. (1995). Factorial learning and the *EM* algorithm. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems 7*, pp. 617–624. Cambridge, MA: MIT Press.

Heckerman, D. (1995.). *A tutorial on learning Bayesian networks.* (Technical Report MSR-TR-95-06). Redmond, WA: Microsoft Research.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations.* Cambridge, MA: MIT Press.

Hinton, G. E., & Zemel, R. S. (1994). Autoencoders, minimum description length, and Helmholtz free energy. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems 6*, pp. 3–10. San Francisco, CA: Morgan Kaufmann.

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

Jordan, M. I., Ghahramani, Z., & Saul, L. K. (1997). Hidden Markov decision trees. In M. Mozer, M. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems 9*. Cambridge, MA: MIT Press.

Jordan, M. I., & Jacobs, R. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181–214.

Kanazawa, K., Koller, D., & Russell, S. J. (1995). Stochastic simulation algorithms for dynamic probabilistic networks. In P. Besnard, , & S. Hanks (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Eleventh Conference.* (pp. 346–351). San Francisco, CA: Morgan Kaufmann.

Krogh, A., Brown, M., Mian, I. S., Sjölander, K., & Haussler, D. (1994). Hidden Markov models in computational biology: Applications to protein modeling . *Journal of Molecular Biology*, *235*, 1501–1531.

Lauritzen, S. L., & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 157–224.

McCullagh, P., & Nelder, J. (1989). *Generalized linear models.* London: Chapman & Hall.

Meila, M., & Jordan, M. I. (1996). Learning fine motion by Markov mixtures of experts. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems 8*, pp. 1003–1009. Cambridge, MA: MIT Press.

Merz, C. J., & Murphy, P. M. (1996). *UCI Repository of machine learning databases [http://www.ics.uci.edu/ mlearn/MLRepository.html].* Irvine, CA: University of California, Department of Information and Computer Science.

Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, *56*, 71–113.

Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods* (Technical Report CRG-TR-93-1). Toronto, Ontario: University of Toronto, Department of Computer Science.

Neal, R. M., & Hinton, G. E. (1993). *A new view of the EM algorithm that justifies incremental and other variants.* Unpublished manuscript, Department of Computer Science, University of Toronto, Ontario.

Parisi, G. (1988). *Statistical field theory.* Redwood City, CA: Addison-Wesley.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* San Mateo, CA: Morgan Kaufmann.

Rabiner, L. R., & Juang, B. H. (1986). An Introduction to hidden Markov models. *IEEE Acoustics, Speech & Signal Processing Magazine*, *3*, 4–16.

Saul, L. K., & Jordan, M. I. (1997). Mixed memory Markov models. In D. Madigan , & P. Smyth (Eds.), *Proceedings of the 1997 Conference on Artificial Intelligence and Statistics.* Ft. Lauderdale, FL.

Saul, L., Jaakkola, T., & Jordan, M. I. (1996). Mean Field Theory for Sigmoid Belief Networks. *Journal of Artificial Intelligence Research*, *4*, 61–76.

Saul, L., & Jordan, M. I. (1995). Boltzmann chains and hidden Markov models. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in neural information processing systems 7*, pp. 435–442. Cambridge, MA: MIT Press.

Saul, L., & Jordan, M. I. (1996). Exploiting tractable substructures in Intractable networks. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems 8*, pp. 486–492. Cambridge, MA: MIT Press.

Smyth, P., Heckerman, D., & Jordan, M. I. (1997). Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, *9*, 227–269.

Stolcke, A., & Omohundro, S. (1993). Hidden Markov model induction by Bayesian model merging. In S.J. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in neural information processing systems 5*, pp. 11–18. San Francisco, CA: Morgan Kaufmann.

Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation (with discussion). *Journal of the American Statistical Association*, *82*, 528–550.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions Information Theory*, *IT-13*, 260–269.

Williams, C. K. I., & Hinton, G. E. (1991). Mean field networks that learn to discriminate temporally distorted strings. In D. Touretzky, J. Elman, T. Sejnowski, & G. Hinton (Eds.), *Connectionist models: Proceedings of the 1990 summer school* (pp. 18–22). San Francisco, CA: Morgan Kaufmann.

Zemel, R. S. (1993). *A minimum description length framework for unsupervised learning.* Ph.D. Thesis, Department of Computer Science, University of Toronto, Toronto, Canada.

Received Date
Accepted Date
Final Manuscript Date