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

Abstract

One of the basic probabilistic tools used for time series modeling is the hidden Markov model (HMM). In an HMM, information about the past of the time series is conveyed through a single discrete variable—the hidden state. We present a generalization of HMMs in which this state is factored into multiple state variables and is therefore represented in a distributed manner. Both inference and learning in this model depend critically on computing the posterior probabilities of the hidden state variables given the observations. We present an exact algorithm for inference in this model, and relate it to the Forward–Backward algorithm for HMMs and to algorithms for more general belief networks. 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 mean field theory. We also present a structured approximation in which the the state variables are decoupled, based on which we derive a tractable learning algorithm. Empirical comparisons suggest that these approximations are efficient and accurate alternatives to the exact methods. Finally, we use the structured approximation to model Bach's chorales and show that it outperforms HMMs in capturing the complex temporal patterns in this dataset.

1 Introduction

A problem of fundamental interest to machine learning is time series modeling. Due 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 and Juang, 1986) and computational molecular biology (Baldi et al., 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. This multinomial assumption allows an efficient parameter estimation algorithm to be derived—the Baum-Welch algorithm. However, it 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 2^{30} distinct states. On the other hand an HMM with a *distributed* state representation could achieve the same task with 30 binary units (Williams and Hinton, 1991). This paper addresses the problem of deriving 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 allow the state space to be decomposed into features that naturally decouple the dynamics of the process generating the time series. Second, distributed state representations simplify the task of modeling time series generated by the interaction of multiple independent 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 representation and proposed a solution based on deterministic Boltzmann learning. 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 distributed state representation, resulting in a more efficient learning procedure. Anticipating the results in section 3, this learning algorithm both obviates the need for the two-phase procedure of Boltzmann machines, and has an exact M-step. 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 can be considered a special case of Bayesian belief networks (Pearl, 1988; Lauritzen and Spiegelhalter, 1988). In belief networks, a probabilistic model is represented as a graphical structure, with the nodes representing random variables and the links representing causal influences. The relation between hidden Markov models and Bayesian belief networks has recently been reviewed in Smyth, Heckerman and Jordan (1996) . While exact belief propagation algorithms exist for belief networks (Jensen et al., 1990), such algorithms soon become intractable for densely-connected models such as the ones we consider in this paper. Kanazawa et al. (1995) apply a stochastic simulation algorithm to overcome this difficulty.

The models considered in this paper are also closely related to recent work in unsupervised learning which has focused on the problem of discovering multiple independent causes, or factors, underlying the data (Barlow, 1989; Redlich, 1993; Zemel, 1993; Saund, 1995; Hinton et al., 1995). The motivation behind factorial learning algorithms is that many real world learning problems are best characterized by an interaction of multiple independent causes or factors. The goal of factorial learning is to invert this data generation process, discovering a representation that will both parsimoniously describe the data and reflect its underlying causes.

This goal can be cast as a statistical inference problem leading to the following common recipe for deriving unsupervised learning algorithms. Starting from a generative model of the data, the learning problem consists of estimating the parameters of the model that best fit the data. This fit is generally measured by the likelihood of the data given the parameters, which can then be maximized as a function of the parameters. Bayesian approaches augment this inference process by incorporating a prior distribution on the parameters and demanding that the result of the learning process be a posterior distribution on the parameters. We will limit ourselves to addressing the problem of computing the likelihood, as this already poses some severe computational problems. The interested Bayesian researcher can extend this framework

An alternative and formally equivalent framework for unsupervised learning can be derived using information theory (Cover and Thomas, 1991). The goal of the learner is to communicate the data efficiently to a receiver, thereby producing a compact representation for the data (Zemel, 1993; Hinton and Zemel, 1994). A cost function quantifying the efficiency of this communication process can be derived from the Minimum Description Length (MDL) principle (Rissanen, 1989). Using Shannon's coding theorem, the MDL cost function can be shown to be equal to the posterior probability of the parameters given the data.

The application of MDL principles to the problem of learning a factorial code has resulted in a learning architecture called cooperative vector quantization (CVQ; Zemel, 1993; Hinton and Zemel, 1994; Ghahramani, 1995). In this model a set of vector quantizers, each representing one of the factors, cooperate to reconstruct the observation. Each vector quantizer is a mixture model (Titterington et al., 1985), and CVQ generalizes mixture models by allowing the mixture components to cooperate in modeling the data set. In simple mixture models each data point must be accounted for by a single mixture component. In CVQs each data point is accounted for by a combination of the contributions from many mixture components, one from each vector quantizer. The total probability density modeled by a CVQ is also a mixture model; however this mixture density is assumed to factorize into a product of densities. Thus, the CVQ is a mixture model with distributed representations for the mixture components. The model presented in this paper is a temporal generalization of the CVQ in which the state of each vector quantizer has a Markov transition structure.

The paper is organized as follows. In the next section we define the probabilistic model for factorial HMMs. 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 dataset consisting of the melody lines from a collection of chorales by J. S. Bach. We discuss several generalizations of the probabilistic model in section 5, and conclude in section 6. Where necessary, details of derivations are provided in the Appendix.

2 The Probabilistic Model

We start by describing the hidden Markov model. In a hidden Markov model, a sequence of observations $\{Y_t\}$ where $t = 1, \ldots T$, is modeled by specifying a probabilistic relation between the observations and a hidden state S_t, and a Markov property on that hidden state. Using this Markov property, the joint probability for the sequences

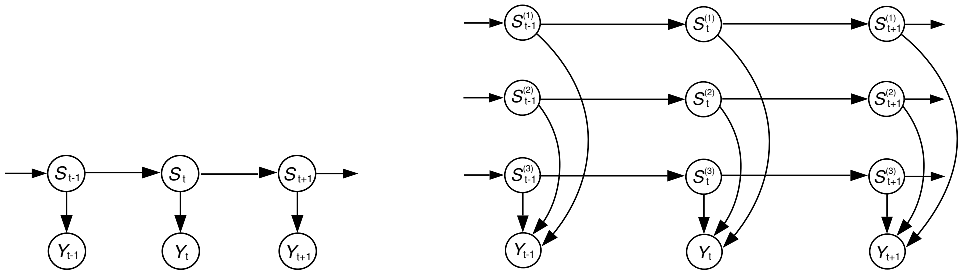

Figure 1: a) A directed acyclic graph (DAG) specifying conditional independence relations for a hidden Markov model. Each node is conditionally independent from its non-descendents given its parents: The output Y_t is conditionally independent from all other variables given the state S_t; and S_t is conditionally independent from $S_1, \ldots, S_{t-2}$ given S_{t-1}. b) A DAG representing the conditional indepence relations in a factorial HMM.

of states S_t and observation Y_t, 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 independences specifies by equation (1) can be expressed graphically in the form of Figure 1a. The state is represented by a single multinomial 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 observables are discrete symbols taking on one of L values, the observation probabilities $P(Y_t|S_t)$ can be fully specified as a $K \times L$ observation matrix. For a continuous observation vector, $P(Y_t|S_t)$ can be modeled in many different forms, such as a Gaussian, mixture of Gaussians, or even a neural network.

In the present paper, we generalize the HMM state representation by allowing the state to 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 $K^{(m)} = K, \forall m$, although all the results we present hold of for the general 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: (1) It is equivalent to an HMM with K^M states. (2) It is unlikely to discover any intersting structure in the K state variables as all variables are allowed to interact arbitrarily. (3) Both the time complexity and sample complexity of the estimation algorithm are exponential in K.

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 transitions 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 1b. The transition structure for this system can be represented as M, $K \times K$ matrices. Generalization to models with some coupling between the state variables will be discussed in section 5.

Note that in Figure 1b, the observation depends on all of the state variables. For continuous observations, one simple form this dependence can take is by assuming that the observation is Gaussian distributed with mean specified by a linear function of the state variables. Representing each state variable as a $K \times 1$ vector which can take on the K discrete values with a 1 in one position and 0s elsewhere,

$$P(Y_t|S_t) \propto \exp\left\{-\frac{1}{2}\left(Y_t - \sum_{m=1}^{M} W^{(m)}S_t^{(m)}\right)' C^{-1} \left(Y_t - \sum_{m=1}^{M} W^{(m)}S_t^{(m)}\right)\right\}. \tag{4}$$

Here, the matrices $W^{(m)}$ are the contributions to the means from each state variable, C is the covariance matrix, and $'$ denotes matrix transpose.

3 Inference and Learning

Given a probabilistic model structure and parameters, the inference problem can generally be stated as computing the posterior probabilities of the hidden variables given the observations. In the context of an HMM used for speech, for example, the observations may be acoustic vectors and the inference problem may be estimating the probabilities for a particular word or sequence of phonemes (the hidden state). This problem can be solved efficiently via the Forward–Backward (F–B) algorithm (Rabiner and Juang, 1986), which can be shown to be equivalent to the Jensen, Lauritzen, and Olesen (JLO; 1990) algorithm for probability propagation in more general belief networks (Smyth et al., 1996). 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 algorithm, which is very closely related to the F-B algorithm and also has analogues in the belief network literature.

The learning problem for probabilistic models consists of two components: learning the structure of the model, and learning the parameters of the model. Structure learning is a topic of current research in both belief network and machine learning communities (e.g. Pearl, 1988; Stolcke and Omohundro, 1993). We concentrate primarily on the problem of learning the parameters for a given structure.

The parameters of a factorial HMM can be estimated via the Expectation Maximization (EM) algorithm (Dempster et al., 1977), which in the case of HMMs is known as the Baum–Welch algorithm (Baum et al., 1970). This procedure iterates between assuming the current parameters to compute posterior probabilities over the hidden states (E-step), and using these probabilities to maximize the expected log likelihood of the parameters (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 starts by defining the expected log likelihood of the complete (observed and hidden) data:

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

where $\mathcal{Q}$ is a function of the parameters of the model ϕ^{new}, given the current estimate of the parameters 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)–(4), we find that $\mathcal{Q}$ can be expressed as a function of three types of expectations over the hidden state variables: $\langle S_t^{(m)} \rangle$, $\langle S_t^{(m)} S_t^{(n)\prime} \rangle$, and $\langle S_{t-1}^{(m)} S_t^{(m)\prime} \rangle$, where $\langle \cdot \rangle$ has been used to abbreviate $E\{\cdot|\phi, \{Y_t\}\}$. Note that in standard HMM notation (Rabiner and Juang, 1986), $\langle S_t^{(m)} \rangle$ corresponds to the γ_t and $\langle S_{t-1}^{(m)} S_t^{(m)\prime} \rangle$ corresponds to the ξ_t computed during the forward-backward iterations, whereas $\langle S_t^{(m)} S_t^{(n)\prime} \rangle$ 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, Dempster et al. (1977) showed that each iteration of the E and M steps increases the likelihood, $P(\{Y_t\}|\phi)$, until convergence to a (local) maximum.

Like in hidden Markov models, the exact M step for factorial HMMs is simple and tractable. For example, the M step for the parameters of the output model described in equation (4) can be found via the normal equations for weighted linear regression. Similarly, the M step for the priors, $\pi^{(m)}$, and state transition matrices, $P^{(m)}$, are identical to those used in the Baum–Welch algorithm. The details of the M-step are described in Appendix A. We now turn to the problem of computing the expectations required for the E-step.

3.1 Exact calculations

Unfortunately, the exact E-step for factorial HMMs is computationally intractable. This fact can best be shown by making reference to the graphical model, although it can also be derived from application of Bayes rule. To compute the posterior probability of a single hidden state node, it is necessary to sum over the probabilities of the *Markov blanket* of that node: the parents, the children, and the parents of the children of that node. In the language of Lauritzen and Spiegelhalter (1988), moralizing and triangulating the graphical structure for a factorial HMM results in a junction tree (actually a chain) with $T(M+1) - M$ cliques of size $M+1$. The resulting belief propagation algorithm has time complexity $O(TMK^{M+1})$ for a single observation sequence of length T. We present a forward-backward type iteration that implements the exact E-step in Appendix B. The naive exact algorithm, consisting of translating the factorial HMM into an equivalent HMM with K^M states and using the F-B algorithm, has time complexity $O(TK^{2M})$. Like in 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—the setting of one state variable only determines the mean of the observation if all the other state variables are fixed.

3.2 Gibbs sampling

Rather than computing the exact posterior probabilities, one can approximate them using a Markov chain 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 ones—Gibbs sampling (Geman and Geman, 1984). For a given observation sequence $\{Y_t\}$, this procedure starts with a random setting of the hidden states $\{S_t\}$. At each time step, 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. We therefore obtain the following sampling scheme in which we only need to examine the states of a few neighboring nodes:

$$S_t^{(m)} \sim P(S_t^{(m)} | \{S_t^{(n)} : n \neq m\}, S_{t-1}^{(m)}, S_{t+1}^{(m)}, Y_t, \phi). \tag{6}$$

These conditional distributions are straightforward to compute—they involve Gaussian distance calculations. Sampling once from each of the TM hidden variables in the model results in a new sample of the overall hidden state of the model and requires $O(TMK)$ operations. The sequence of overall states resulting from each pass of Gibbs sampling define a Markov chain over the state space of the model. Assuming all probabilities are bounded away from zero, this Markov chain is guaranteed to converge to the posterior probabilities of the states given the observations. 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 $\langle S_t^{(m)} \rangle$, $\langle S_t^{(m)} S_t^{(n)\prime} \rangle$ and $\langle S_{t-1}^{(m)} S_t^{(m)\prime} \rangle$ are collected using the states visited and the probabilities estimated during this sampling process. One problem often found with such sampling schemes is that it is very difficult to diagnose convergence.

A natural extension of this sampling procedure for the hidden states is to consider the *parameters* of the model as a separate set of hidden variables. This lends itself to a Bayesian version of EM in which, rather than maximizing the likelihood of the parameters in the M-step, one samples from their posterior distribution given the observations and hidden states. This procedure, which can be seen as a particular form of Gibbs sampling, is also known as data augmentation (Tanner and Wong, 1987).

3.3 Mean field approximation

In this section, we present a second approximation of the posterior probability of the hidden states which is both tractable and deterministic. The basic idea is to approximate the posterior distribution P by a tractable distribution Q. This approximation provides a lower bound on the log likelihood which can be used to obtain an efficient learning algorithm.

The argument can be formalized as follows (Saul et al., 1996). Any approximating distribution $Q(\{S_t\}|\{Y_t\})$ can be used to define a lower bound on the log likelihood:

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

where we have made use of Jensen's inequality (Cover and Thomas, 1991) in the last step. The difference between this bound and the log likelihood is given by the Kullback-Liebler divergence:

$$\mathrm{KL}(Q\|P) = \sum_{\{S_t\}} Q(\{S_t\}|\{Y_t\}) \, \log \left[\frac{Q(\{S_t\}|\{Y_t\})}{P(\{S_t\}|\{Y_t\})} \right]. \tag{7}$$

From here on, we drop $\{Y_t\}$ from the conditioning side of Q for notational convenience.

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—i.e. a graphical representation which 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 (7).

The simplest approximation, in which all the state variables are independent given the observations (Figure 2a), can be written as

$$Q(\{S_t\}|\theta) = \prod_{t=1}^{T} \prod_{m=1}^{M} Q(S_t^{(m)}|\theta_t^{(m)}). \tag{8}$$

The parameters of this distribution, $\theta = \{\theta_t^{(m)}\}$, are the means of the state variables:

$$Q(S_t^{(m)}|\theta_t^{(m)}) = \theta_t^{(m)}. \tag{9}$$

In the statistical mechanics literature, this completely factorized approximation is known as a mean field approximation (Parisi, 1988). The approximation is made as tight as possible by varying θ so as to minimize the KL divergence. Taking the derivatives of (7) with respect to $\theta_t^{(m)}$ and setting to zero, we obtain the set of fixed

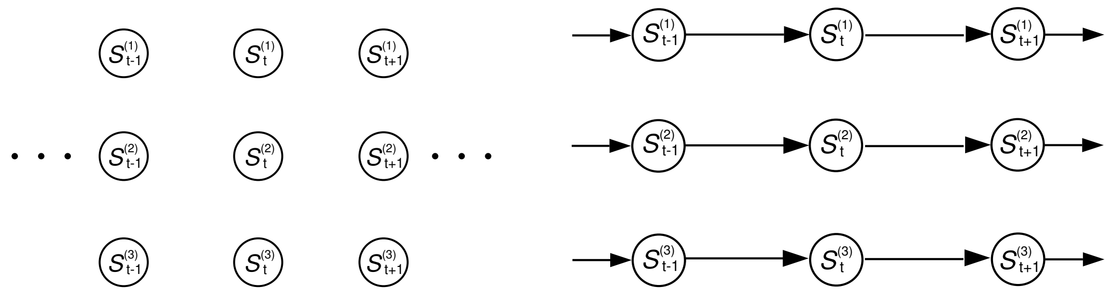

Figure 2: a) The mean field approximation assumes that all the state variables are independent (conditional on the observation sequence). b) The structured approximation assumes that the state variables retain their Markov structure within each chain, but are independent across chains.

point equations defined by (see Appendix C):

$$\begin{aligned}\theta_t^{(m)\,\text{new}} &= \varphi\left\{W^{(m)\prime}C^{-1}\left(Y_t - \hat{Y}_t\right) + W^{(m)\prime}C^{-1}W^{(m)}\theta_t^{(m)} - \frac{1}{2}\Delta^{(m)} + \right.\\ &\qquad \left. + (\log P^{(m)})\,\theta_{t-1}^{(m)} + (\log P^{(m)})'\,\theta_{t+1}^{(m)}\right\} \qquad (10)\end{aligned}$$

where

$$\hat{Y}^t = \sum_{\ell=1}^{M} W^{(\ell)}\theta_t^{(\ell)}, \qquad (11)$$

$\Delta^{(m)}$ is the vector of diagonal elements of $W^{(m)\prime}C^{-1}W^{(m)}$, and $\varphi\{\cdot\}$ is the softmax nonlinearity:

$$\varphi\{X\} = \frac{\exp\{X_i\}}{\sum_i \exp\{X_i\}},$$

normalized over the means associated with each state vector. The first term of (10) 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 mean field parameter. The next two terms arise from the fact that the second order correlation $\langle S_t^{(m)} S_t^{(m)} \rangle$ evaluated under the mean field distribution is equal to the mean $\theta_t^{(m)}$. The last two terms introduce dependencies forwards and backwards in time. We therefore see that although the posterior is approximated with a completely factorized distribution, the mean field equations couple the parameter for each node with the parameters of its Markov blanket. In this sense, the mean field equations propagate information along the same directions as the exact algorithms for probability propagation.

Each hidden state vector is updated in turn using (10), with a time complexity of $O(TKM)$ per iteration. Convergence is diagnosed by monitoring the KL divergence in the mean field distribution between successive time steps; in practice convergence is very rapid (about 2 to 10 iterations of (10)). Once the mean field equations have converged, the expectations required for the E-step can be obtained as a simple function of the parameters (equations (40)–(42) in Appendix C).

3.4 Structured approximation

The mean field approximation presented in the previous section factorized the posterior probability such that all the state variables were independent. In contrast to this rather extreme factorization, in this section we present an approximation which is both tractable and preserves much of the structure of the original system. The factorial HMM is approximated by M uncoupled HMMs (Figure 2b). Within each HMM, efficient and exact inference is implemented via the forward–backward algorithm. The approach of exploiting 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, this structured approximation also provides a lower bound on the log likelihood and can be used to obtain an efficient learning algorithm.

We write the structured approximation as

$$Q(\{S_t\}|\theta) = \prod_{m=1}^{M} Q(S_1^{(m)}|\theta) \prod_{t=2}^{T} Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta), \tag{12}$$

where

$$\begin{aligned} Q(S_1^{(m)}|\theta) &= \pi^{(m)} h_1^{(m)} && (13) \\ Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta) &= P(S_t^{(m)}|S_{t-1}^{(m)}, \phi)\, h_t^{(m)} = P^{(m)} h_t^{(m)}. && (14) \end{aligned}$$

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 to each state variable. Comparing equations (12)–(14) to equation (1), we note that $h_t^{(m)}$ has taken the place of $P(Y_t|S_t)$.

Applying the same arguments as before, we obtain a set of fixed point equations for $h_t^{(m)}$ that minimize the KL divergence between Q and P (see Appendix D):

$$h_t^{(m)\,\text{new}} \propto \exp\left\{ W^{(m)\prime} C^{-1} \left(Y_t - \hat{Y}_t \right) + W^{(m)\prime} C^{-1} W^{(m)} \langle S_t^{(m)} \rangle - \frac{1}{2} \Delta^{(m)} \right\}, \tag{15}$$

where

$$\hat{Y}^t = \sum_{\ell=1}^{M} W^{(\ell)} \langle S_t^{(\ell)} \rangle, \tag{16}$$

and $\Delta^{(m)}$ is defined as before. The parameter $h_t^{(m)}$ obtained from these fixed point equations is the input to state variable $S_t^{(m)}$ in hidden Markov model m. Using these inputs, the F–B algorithm is used to compute a new set of expectations for $\langle S_t^{(m)} \rangle$, which are fed back into (15) and (16). The F–B algorithm is therefore used as a subroutine in the minimization of the KL divergence.

Note the similarity between equations (15)–(16) and equations (10)–(11) 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 mean field parameters. In the structured approximation, the dependence of $\langle S_t^{(m)} \rangle$ on $h_t^{(m)}$ is computed via the F–B algorithm. Note also that (15) does not contain terms involving the prior, $\pi^{(m)}$, or transition matrix, $P^{(m)}$. These terms have cancelled out by our choice of approximation.

4 Simulations

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 performance, as measured by the likelihood of the model obtained. The second experiment sought to determine whether the decomposition of the state space in factorial HMMs presented any advantage in modeling a real time series dataset which we might assume to have complex internal structure—Bach's chorale melodies.

4.1 Experiment 1: Performance and Timing Benchmark

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

- HMM trained using Baum-Welch.
- Factorial HMM trained using exact inference for the E-step. The exact algorithm used was 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. The number of samples was fixed at 10, a priori.
- Factorial HMM trained using the mean field approximation.
- Factorial HMM trained using the structured approximation.

All factorial HMMs had a structure consisting of M underlying Markov models with K states each; 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 set consisted of 10 sequences of length 20 where the observable was a 4-dimensional vector; the test set was 20 such sequences. Algorithms were run until convergence, monitored by relative change in log likelihood, or a maximum of 100 cycles of EM. At the end of learning, the log likelihoods on the training and test set were computed for all models using the exact algorithm.

M	K	Alg	Train		Test	
3	2	HMM	649	$\pm$ 8	358	$\pm$ 81
		Exact	877	$\pm$ 0	768	$\pm$ 0
		Gibbs	710	$\pm$ 152	627	$\pm$ 129
		MF	755	$\pm$ 168	670	$\pm$ 137
		Struct	755	$\pm$ 168	661	$\pm$ 152
3	3	HMM	670	$\pm$ 26	-782	$\pm$ 128
		Exact	568	$\pm$ 164	276	$\pm$ 62
		Gibbs	564	$\pm$ 160	305	$\pm$ 51
		MF	495	$\pm$ 83	326	$\pm$ 62
		Struct	606	$\pm$ 154	317	$\pm$ 47
5	2	HMM	588	$\pm$ 37	-2634	$\pm$ 566
		Exact	223	$\pm$ 76	159	$\pm$ 80
		Gibbs	123	$\pm$ 103	73	$\pm$ 95
		MF	292	$\pm$ 101	237	$\pm$ 103
		Struct	242	$\pm$ 247	184	$\pm$ 250
5	3	HMM	1610	$\pm$ 102	$-\infty^*$	
		Exact	-283	$\pm$ 130	-331	$\pm$ 117
		Gibbs	-206	$\pm$ 69	-299	$\pm$ 79
		MF	-290	$\pm$ 5	-368	± 4
		Struct	-275	$\pm$ 22	-368	$\pm$ 8

Table 1: Comparison of factorial HMM on four problems of varying size. The Train and Test columns show the log likelihood $\pm$ one standard deviation on the two datasets.

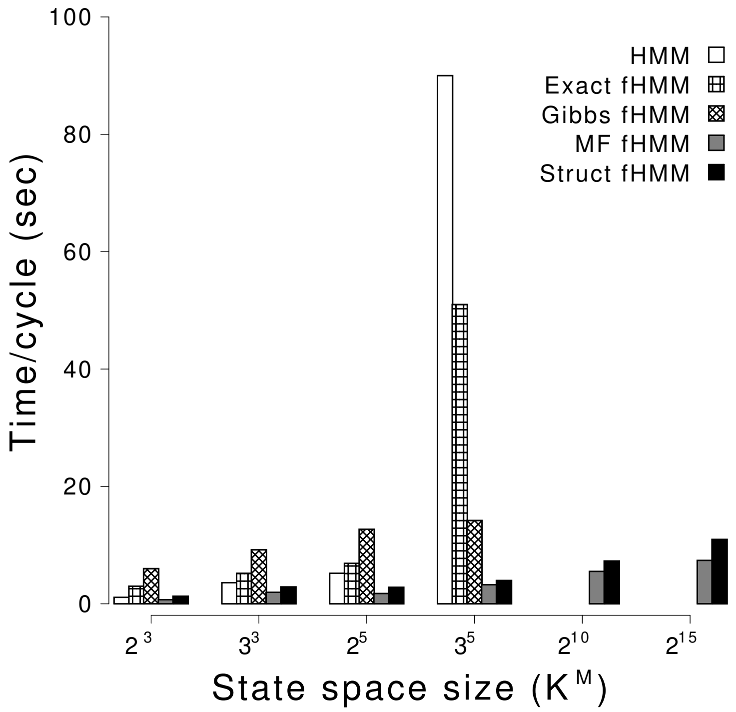

Figure 3: Time per cycle of EM on a Silicon Graphics R4400 processor running Matlab.

The results indicate that even for moderately large state spaces ($M \geq 3$ and $K \geq 3$), the standard HMM with K^M states suffered from severe overfitting (Table 4.1). Furthermore, both the standard HMM and the exact E-step factorial HMM are extremely slow on the larger problems (Figure 3). The Gibbs sampling, mean field, and structured approximations offered roughly comparable performance at a great increase in speed.

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 learning the structure of a collection of musical pieces.

The dataset consisted of discrete event sequences encoding the melody lines of J. S. Bach's Chorales (obtained from the UCI Repository for Machine Learning (Murphy and Aha, 1992); originally discussed in Conklin and Witten, 1995). Each event in the sequence consisted of the setting of 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

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)

Table 2: Attributes in the Bach chorale dataset. The key signature and time signature attributes were constant over the duration of the chorale.

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 approximate the E-step for factorial HMMs we used the structured 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 mean field approximation since more of the dependencies of the original model are preserved.

The test set log likelihoods for the HMMs, shown in Figure 4a, exhibited the typical U-shaped curve demonstrating a trade-off between bias and variance (Geman et al., 1992). HMMs with fewer than 10 states did not capture much of the structure of the chorales, 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 was -6.26 nats, obtained by an HMM with 30 hidden states.

Almost half (31/63) of the runs of the factorial HMM surpassed the best test set log likelihood of the HMM (Figure 4b and c). Furthermore, the best factorial HMMs had very large state spaces, approaching 1000 states, which would result in severe overfitting and prohibitively slow training for unconstrained HMMs. It is possible to constrain HMMs to have many states with few parameters by allowing only a limited number of transitions. However, such HMMs do not have the componential state representation of factorial HMMs. This componentiality, which allows for several dynamic processes to be represented at the same time, is most likely to account for the superior performance of factorial HMMs in modeling Bach's chorales.

5 Generalizations

5.1 Discrete Observables

The probabilistic model presented in this paper assumed real-valued Gaussian observations. Although this was not emphasized, one of the advantages arising from this assumption is that the conditional density of a D-dimensional observation $P(Y_t|S_t^{(1)}, \ldots, S_t^{(M)})$

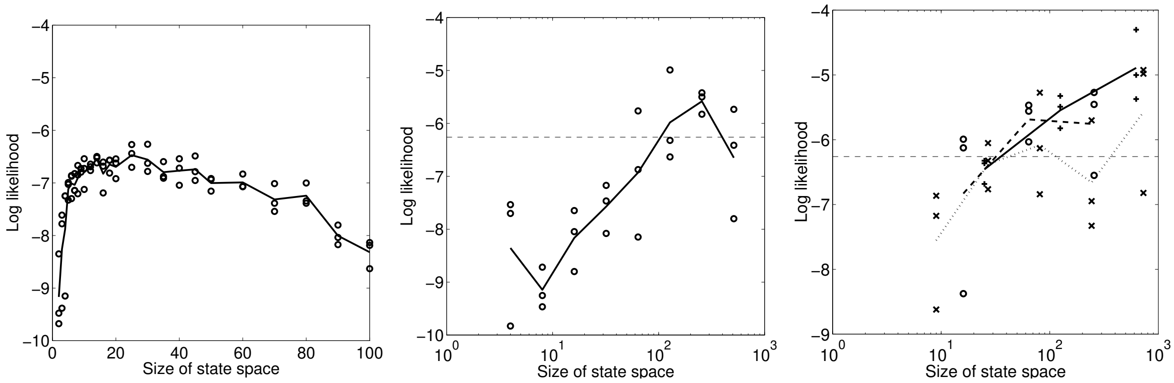

Figure 4: Test set log likelihood per event of the Bach chorale dataset as a function of number of states for a) HMMs, b) factorial HMMs with $K = 2$ and c) factorial HMMs with $K = 3$ ($\times$'s; dotted line), $K = 4$ ($\circ$'s; heavy dashed line) and $K = 5$ ($+$'s; solid line). Each circle represents a single run; the lines indicate the mean performances. The thin dashed line in b and c indicates the log likelihood 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 same algorithm.

can be compactly specified through M $D \times K$ mean matrices, and one $D \times D$ covariance matrix. Furthermore, the M-step for such a model reduces to a set of regression 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 belief networks 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 $D > 2$,

$$P(Y_t|S_t^{(1)}, \ldots, S_t^{(M)}) \propto \exp\left\{\sum_{m=1}^{M} W^{(m)} S_t^{(m)}\right\}. \tag{17}$$

Like the Gaussian model, this specification is again compact, using M $D \times K$ matrices. (Note that also like the Gaussian model, the means of the $W^{(m)}$ are overparametrized, see Appendix A.) Unfortunately, the nonlinearity induced by the softmax function makes both the E-step and M-step of the algorithm more difficult (see Neal, 1992; Hinton et al., 1995; and Saul et al, 1996, for discussions of different approaches to learning in sigmoid belief 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. 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

$$P(S_t|S_{t-1}) = P(S_t^{(1)}|S_{t-1}^{(1)}) \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}, S_{t-1}^{(m-1)}). \tag{18}$$

Similar exact, mean field, and Gibbs sampling procedures can be defined for this architecture. Note however, 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.[1] In the graphical model formalism these correspond to symmetric undirected links, making the model a chain graph. While the JLO 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. Like in Boltzmann machines (Ackley et al., 1985), 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 domain, some of the observations can be better thought of as inputs or predictor 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 which are conditioned on inputs have been recently presented in the literature (Cacciatore and Nowlan, 1994; Bengio and Frasconi, 1995; Meila and Jordan, 1996). Given a sequence of input vectors $\{X_t\}$, the probabilistic model for an input-conditioned factorial HMM is

$$\begin{aligned} P(\{S_t, Y_t\}|\{X_t\}) &= \prod_{m=1}^{M} P(S_1^{(m)}|X_1) P(Y_1|S_1^{(m)}, X_1) \\ &\quad \cdot \prod_{t=2}^{T} \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}, X_t) P(Y_t|S_t^{(m)}, X_t). \end{aligned} \tag{19}$$

[1] This is analogous to the fully-connected Boltzmann machine with N units (Ackley et al., 1985), 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.

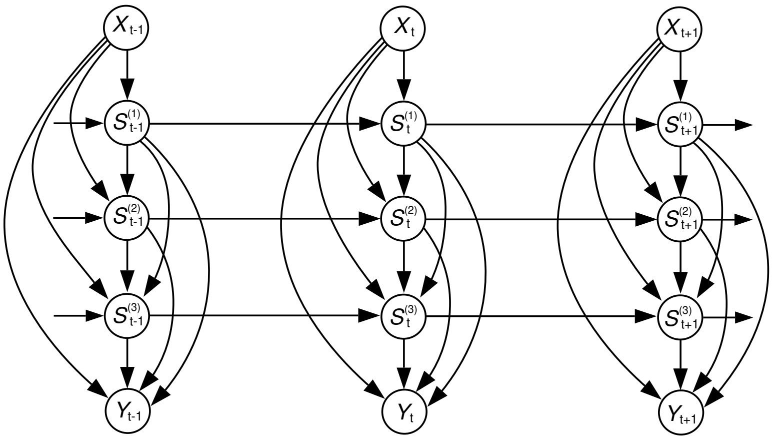

Figure 5: The hidden Markov decision tree.

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 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 mean field and structured approximations can also be generalized simply as long as 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 (IRLS; McCullagh and Nelder, 1983).

5.4 Hidden Markov Decision Trees

An interesting generalization of factorial HMMs results if we condition on an input X_t and order the M state variables such that $S_t^{(m)}$ depends on $S_t^{(n)}$ for $1 \leq n < m$ (Figure 5). The resulting architecture can be seen as a hidden Markov decision tree (HMDT). Consider how this probabilistic 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 induces a stochastic partitioning of 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 based on the input X_1 and the M K-way decisions at the 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 decision tree is dependent on the decision taken at that node in the previous time step. Thus, the "hierarchical mixture of experts" architecture (Jordan and Jacobs, 1994) is generalized such that it includes 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 (1996).

6 Conclusion

The basic contribution of this paper is a learning algorithm for hidden Markov models with distributed state representations. The standard Baum-Welch procedure is intractable for such architectures as the size of the state space generated from the cross product of M K-valued features is $O(K^M)$. The time complexity of Baum-Welch is quadratic in this size. More importantly, unless special constraints are applied to this cross-product HMM architecture, the number of parameters also grows as $O(K^{2M})$, which can result in severe overfitting.

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, and solving. We start by expanding $\mathcal{Q}$ using equations (1)–(4):

$$\begin{aligned}\mathcal{Q} \;=\; & -\frac{1}{2}\sum_{t=1}^{T}\left[Y_t' C^{-1} Y_t - 2\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} \langle S_t^{(m)}\rangle + \sum_{m=1}^{M}\sum_{n=1}^{M} tr\{W^{(m)'} C^{-1} W^{(n)} \langle S_t^{(n)} S_t^{(m)'}\rangle\}\right] \\ & + \sum_{m=1}^{M} \langle S_1^{(m)'}\rangle \log \pi^{(m)} + \sum_{t=2}^{T}\sum_{m=1}^{M} tr\left\{(\log P^{(m)}) \langle S_{t-1}^{(m)} S_t^{(m)'}\rangle\right\} - \log Z, \end{aligned} \tag{20}$$

where 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 $W^{(m)}$:

$$\frac{\partial \mathcal{Q}}{\partial W^{(m)}} = \sum_{t=1}^{T}\left[\sum_{n=1}^{M} W^{(n)} \langle S_t^{(n)} S_t^{(m)'}\rangle - Y_t \langle S_t^{(m)'}\rangle\right] = 0. \tag{21}$$

Assuming Y_t is a $D \times 1$ vector, let S_t be the $MK \times 1$ vector obtained by concatenating the $S^{(m)}$, and W be the $D \times MK$ matrix obtained by concatenating the $D \times K$ $W^{(m)}$

matrices. Then,

$$W^{\text{new}} = \left(\sum_{t=1}^{T} Y_t \langle S_t' \rangle\right) \left(\sum_{t=1}^{T} \langle S_t S_t' \rangle\right)^{\dagger}. \tag{22}$$

where † denotes the Moore-Penrose pseudo-inverse. Note that the model is slightly overparametrized since the $D \times 1$ means of each of the $W^{(m)}$ matrices add up to a single mean. Using the pseudo-inverse obviates the need to explicitly subtracted 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

$$\pi^{(m)\,\text{new}} = \langle S_1^{(m)} \rangle. \tag{23}$$

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 $P^{(m)}$:

$$P_{i,j}^{(m)\,\text{new}} = \frac{\sum_{t=1}^{T} \langle S_{t,i}^{(m)} S_{t-1,j}^{(m)} \rangle}{\sum_{t=1}^{T} \langle S_{t-1,j}^{(m)} \rangle}. \tag{24}$$

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

$$\frac{\partial \mathcal{Q}}{\partial C^{-1}} = \frac{1}{2} C + \sum_{t=1}^{T} \left[\sum_{m=1}^{M} Y_t \langle S_t^{(m)\prime} \rangle W^{(m)\prime} - \frac{1}{2} Y_t Y_t' - \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{M} W^{(n)} \langle S_t^{(n)} S_t^{(m)\prime} \rangle W^{(m)\prime} \right], \tag{25}$$

where the first term arises from the normalization term for the Gaussian density function. Substituting (21) and re-organizing we get

$$C^{\text{new}} = \sum_{t=1}^{T} Y_t Y_t' - \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle \, Y_t'. \tag{26}$$

Note that, 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.

B Exact Forward–Backward Algorithm

The following algorithm defines 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 F–B 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 $K \times K$ transition matrices.

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

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

Then we obtain the following forward recursions:

$$\alpha_t = P(Y_t|S_t^{(1)}, \ldots, S_t^{(M)}, \phi)\alpha_t^{(0)} \tag{27}$$

$$\alpha_t^{(m-1)} = \sum_{S_{t-1}^{(m)}} P(S_t^{(m)}|S_{t-1}^{(m)})\alpha_t^{(m)}. \tag{28}$$

Note that at the end of the forward recursions, the likelihood of the observation sequence is the sum of the elements in α_T.

Similarly, to obtain the backward recursions we define:

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

Then:

$$\beta_{t-1}^{(M)} = P(Y_t|S_t^{(1)}, \ldots, S_t^{(M)}, \phi)\beta_t \tag{29}$$

$$\beta_{t-1}^{(m-1)} = \sum_{S_t^{(m)}} P(S_t^{(m)}|S_{t-1}^{(m)})\beta_{t-1}^{(m)}. \tag{30}$$

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

$$\gamma_t = P(S_t|\{Y_\tau\}_1^T, \phi) = \frac{\alpha_t\beta_t}{\sum_{S_t}\alpha_t\beta_t}. \tag{31}$$

This algorithm can be shown to be equivalent to the JLO algorithm for probability propagation in belief networks. 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

$$\begin{aligned}
\langle S_t^{(m)} \rangle &= \sum_{S_t^{(n)}\,(n \neq m)} \gamma_t && (32)\\
\langle S_t^{(m)} S_t^{(n)\prime} \rangle &= \sum_{S_t^{(r)}\,(r \neq m \wedge r \neq n)} \gamma_t && (33)\\
\langle S_{t-1}^{(m)} S_t^{(m)\prime} \rangle &= \frac{\displaystyle\sum_{S_{t-1}^{(n)}, S_t^{(r)}\,(n \neq m \wedge r \neq m)} \alpha_{t-1} P(S_t|S_{t-1}) P(Y_t|S_t) \beta_t}{\displaystyle\sum_{S_{t-1}^{(n)}, S_t^{(r)}} \alpha_{t-1} P(S_t|S_{t-1}) P(Y_t|S_t) \beta_t}. && (34)
\end{aligned}$$

C Mean field approximation

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

$$P(\{S_t^{(m)}\}|\{Y_t\}, \phi) = \frac{1}{Z} \exp\{-H(\{S_t, Y_t\})\}, \tag{35}$$

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

$$\begin{aligned}
H(\{S_t, Y_t\}) &= \frac{1}{2} \sum_{t=1}^{T} \left(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \right)' C^{-1} \left(Y_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \right) \\
&\quad - \sum_{m=1}^{M} S_1^{(m)\prime} \log \pi^{(m)} - \sum_{t=2}^{T} \sum_{m=1}^{M} S_t^{(m)\prime} (\log P^{(m)}) S_{t-1}^{(m)}. && (36)
\end{aligned}$$

Similarly, the probability distribution given by the mean field approximation (8)–(9) can be written as

$$Q(\{S_t\}|\theta) = \frac{1}{Z_Q} \exp\{-H_Q(\{S_t\})\}, \tag{37}$$

where

$$H_Q(\{S_t\}) = -\sum_{t=1}^{T} \sum_{m=1}^{M} S_t^{(m)\prime} \log \theta_t^{(m)}. \tag{38}$$

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

$$\mathrm{KL}(Q\|P) = \langle H \rangle - \langle H_Q \rangle - \log Z_Q + \log Z. \tag{39}$$

The following facts can be verified from the definition of the mean field approximation:

$$\begin{aligned}
\langle S_t^{(m)} \rangle &= \theta_t^{(m)} && (40)\\
\langle S_{t-1}^{(m)} S_t^{(m)\prime} \rangle &= \theta_t^{(m)} \theta_{t-1}^{(m)\prime} && (41)\\
\langle S_t^{(m)} S_t^{(n)\prime} \rangle &= \begin{cases} \theta_t^{(m)} \theta_t^{(n)\prime} & \text{if } m \neq n \\ diag\{\theta_t^{(m)}\} & \text{if } m = n, \end{cases} && (42)
\end{aligned}$$

where *diag* is an operator that takes a vector and returns a square matrix with the elements of the vector along its diagonal, and 0s everywhere else. The KL divergence can therefore be expanded out as

$$\begin{aligned}
\mathrm{KL} &= \sum_{t=1}^{T}\sum_{m=1}^{M} \theta_t^{(m)\prime} \log \theta_t^{(m)} + \frac{1}{2}\sum_{t=1}^{T}\Bigg[Y_t' C^{-1} Y_t - 2\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} \theta_t^{(m)} \\
&\quad + \sum_{m=1}^{M}\sum_{n\neq m}^{M} tr\{W^{(m)\prime} C^{-1} W^{(n)} \theta_t^{(n)} \theta_t^{(m)\prime}\} + \sum_{m=1}^{M} tr\{W^{(m)\prime} C^{-1} W^{(m)} \theta_t^{(m)\prime}\}\Bigg] \\
&\quad + \sum_{m=1}^{M} \theta_1^{(m)\prime} \log \pi^{(m)} + \sum_{t=2}^{T}\sum_{m=1}^{M} tr\{\theta_{t-1}^{(m)} \theta_t^{(m)\prime} \log P^{(m)}\} - \log Z_Q + \log Z, \quad (43)
\end{aligned}$$

where *tr* is the trace operator for square matrices. Taking derivatives with respect to $\theta_t^{(m)}$, we obtain

$$\begin{aligned}
\frac{\partial \mathrm{KL}}{\partial \theta_t^{(m)}} &= \log \theta_t^{(m)} - Y_t' C^{-1} W^{(m)} + \sum_{n\neq m}^{M} \theta_t^{(n)\prime} W^{(n)\prime} C^{-1} W^{(m)} + \frac{1}{2} diag\{W^{(m)\prime} C^{-1} W^{(m)}\} \\
&\quad -(\log P^{(m)})\, \theta_{t-1}^{(m)} - (\log P^{(m)})'\, \theta_{t+1}^{(m)} + c, \quad (44)
\end{aligned}$$

where c is a constant arising from $\log Z_Q$, ensuring that the $\theta_t^{(m)}$ are normalized. Setting this derivative equal to 0 and solving for $\theta_t^{(m)}$ gives equation (10).

D Structured approximation

For the structured approximation, H_Q is defined as

$$H_Q(\{S_t\}) = -\sum_{m=1}^{M} S_1^{(m)\prime} \log \pi^{(m)} - \sum_{t=2}^{T}\sum_{m=1}^{M} S_t^{(m)\prime} (\log P^{(m)}) S_{t-1}^{(m)} - \sum_{t=1}^{T}\sum_{m=1}^{M} S_t^{(m)\prime} \log h_t^{(m)}. \quad (45)$$

Using (36) we write the KL divergence as

$$\begin{aligned}
\mathrm{KL} &= \sum_{t=1}^{T}\sum_{m=1}^{M} \langle S_t^{(m)}\rangle \log h_t^{(m)} + \frac{1}{2}\sum_{t=1}^{T}\Bigg[Y_t' C^{-1} Y_t - 2\sum_{m=1}^{M} Y_t' C^{-1} W^{(m)} \langle S_t^{(m)}\rangle \\
&\quad + \sum_{m=1}^{M}\sum_{n\neq m}^{M} tr\{W^{(m)\prime} C^{-1} W^{(n)} \langle S_t^{(n)}\rangle \langle S_t^{(m)\prime}\rangle\} + \sum_{m=1}^{M} tr\{W^{(m)\prime} C^{-1} W^{(m)} \langle S_t^{(m)\prime}\rangle\}\Bigg] \\
&\quad - \log Z_Q + \log Z. \quad (46)
\end{aligned}$$

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 $\log h_\tau^{(n)}$,

$$\begin{aligned}
\frac{\partial \mathrm{KL}}{\partial \log h_\tau^{(n)}} &= \langle S_\tau^{(n)}\rangle + \sum_{t=1}^{T}\sum_{m=1}^{M}\Bigg[\log h_t^{(m)} - Y_t' C^{-1} W^{(m)} + \sum_{\ell\neq m}^{M} \langle S_t^{(\ell)}\rangle W^{(\ell)\prime} C^{-1} W^{(m)} \\
&\quad + \frac{1}{2} diag\{W^{(m)\prime} C^{-1} W^{(m)}\}\Bigg] \frac{\partial \langle S_t^{(m)}\rangle}{\partial \log h_\tau^{(n)}} - \langle S_\tau^{(n)}\rangle. \quad (47)
\end{aligned}$$

The last term was obtained by making use of the fact that

$$\frac{\partial \log Z_Q}{\partial \log h_\tau^{(n)}} = \langle S_\tau^{(n)} \rangle, \tag{48}$$

and cancels out the first term. Setting the terms inside the brackets in (47) equal to zero yields equation (15).

Acknowledgements

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 grant from Siemens Corporation, and by grant N00014-94-1-0777 from the Office of Naval Research.

References

Ackley, D., Hinton, G., and Sejnowski, T. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169.

Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M. (1994). Hidden Markov models of biological primary sequence information. *Proc. Nat. Acad. Sci. (USA)*, 91(3):1059–1063.

Barlow, H. (1989). Unsupervised learning. *Neural Computation*, 1:295–311.

Baum, L., Petrie, T., Soules, G., and 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. and Frasconi, P. (1995). An input–output HMM architecture. In Tesauro, G., Touretzky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems 7*, pages 427–434. MIT Press, Cambridge, MA.

Cacciatore, T. W. and Nowlan, S. J. (1994). Mixtures of controllers for jump Linear and non-linear plants. In Cowan, J. D., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems 6*, pages 719–726. Morgan Kaufman Publishers, San Francisco, CA.

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

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. Wiley, New York.

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

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

Geman, S. and 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 Tesauro, G., Touretzky, D., and Leen, T., editors, *Advances in Neural Information Processing Systems 7*. MIT Press, Cambridge, MA.

Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161.

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

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

Jordan, M. I., Ghahramani, Z., and Saul, L. K. (1996). Hidden Markov decision trees. *Submitted to NIPS 8*.

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

Kanazawa, K., Koller, D., and Russell, S. J. (1995). Stochastic simulation algorithms for dynamic probabilistic networks. In *Proceedings of the 11th Annual Conference on Uncertainty in AI*, pages 346–351.

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

Meila, M. and Jordan, M. I. (1996). Learning fine motion by Markov mixtures of experts. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems 8*. MIT Press.

Murphy, P. M. and Aha, D. W. (1992). *UCI Repository of machine learning databases [Machine-readable data repository]*. University of California, Department of Information and Computer Science, Irvine, CA.

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. *University of Toronto, Technical Report CRG-TR-93-1.*

Parisi, G. (1988). *Statistical Field Theory.* Addison-Wesley, Redwood City, CA.

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

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

Redlich, A. N. (1993). Supervised factorial learning. *Neural Computation*, 5:750–766.

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry.* World Scientific, Singapore.

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

Saul, L. and Jordan, M. I. (1995). Boltzmann chains and hidden Markov models. In Tesauro, G., Touretzky, D., and Leen, T., editors, *Advances in Neural Information Processing Systems 7.* MIT Press, Cambridge, MA.

Saul, L. and Jordan, M. I. (1996). Exploiting tractable substructures in Intractable networks. In Touretzky, D., Mozer, M., and Hasselmo, M., editors, *Advances in Neural Information Processing Systems 8.* MIT Press.

Saund, E. (1995). A multiple cause mixture model for unsupervised learning. *Neural Computation*, 7(1):51–71.

Smyth, P., Heckerman, D., and Jordan, M. I. (1996.). Probabilistic independence networks for hidden Markov probability models. *Microsoft Research Technical Report MSR-TR-96-03.*

Stolcke, A. and Omohundro, S. (1993). Hidden Markov model induction by Bayesian model merging. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing Systems 5*, pages 11–18. Morgan Kaufmann, San Francisco, CA.

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

Titterington, D. M., Smith, A., and Makov, U. (1985). *Statistical analysis of finite mixture distributions.* Wiley, Chichester.

Williams, C. and Hinton, G. (1991). Mean field networks that learn to discriminate temporally distorted strings. In Touretzky, D., Elman, J., Sejnowski, T., and Hinton, G., editors, *Connectionist Models: Proceedings of the 1990 Summer School*, pages 18–22. Morgan Kaufmann Publishers, Man Mateo, CA.

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