

A New Formulation of Coupled Hidden Markov Models

Shi Zhong and Joydeep Ghosh
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084, USA

June, 2001

Abstract

Among many variations of more complex hidden Markov models, coupled hidden Markov models (CHMM) have recently attracted increased interests in many practical applications. This paper describes a new CHMM formulation in which the joint transition probability is modeled as a linear combination of the marginal transition probabilities and the weights used to capture the interactions among multiple HMMs. The new formulation greatly reduces the parameter space for CHMM. New approximated forward procedure and training algorithm are proposed to reduce the computational complexity to a practical level. Experimental results show that our new CHMM formulation perform better in our recognition task on both artificial and real data compared to non-coupled HMMs. And our approximated training algorithm still converges to local maxima even though there is no theoretical proof thus provides an efficient practical algorithm for our CHMM formulation.

1 Introduction

Hidden Markov models (HMM) have been heavily researched and used for the past several decades, especially in the speech recognition area[6, 22]. A standard HMM model uses a hidden state at time t to summarize all the information it has before t and thus the observation at time t depends only on the hidden state at time t . And the hidden state sequence over time in an HMM model is a Markov chain. In this paper we consider only the first order HMMs, in which the hidden state sequence is a first order Markov chain. Such an HMM unrolled over several time slices is shown in Fig. 1.

Basically, there are three problems of interest in a standard HMM model. Problem 1, also called *observation evaluation* problem, is on how to compute the probability of an observation sequence O given a model λ , i.e. $P(O|\lambda)$. Problem 2 is on how to find the 'optimal' state sequence given an observation sequence and the model that generated the observation. One optimal solution, for instance, is $\arg \max_S P(S|O, \lambda)$. Problem 3, also the most important and difficult problem, is on how to choose an 'optimal' set of parameters for the model given some observation sequences. Usually, the parameters are adjusted iteratively to maximize $P(O|\lambda)$. In this paper, we mainly discuss solutions to problem 1 and problem 3 simply because that problem 2 is often computationally prohibitive and less meaningful for our coupled HMM models. Problem 1 is solved by the well-known forward/backward procedures. The main training algorithm used to solve problem 3 for standard HMM models is the Baum-Welch algorithm, or equivalently, the Expectation-Maximization (EM) algorithm. The algorithm was first described and proved to converge by Baum and his colleagues[3, 4, 5]. The details are discussed in next several sections.

Here we want to point out some problems or limitations associated with the standard HMM model. First, the objective function $P(O|\lambda)$ that the learning algorithms try to optimize is separate for each single HMM model. More complex objective functions (e.g. mutual information objective[1]) have been proposed but can not take advantage of the fast EM algorithm. Second, the separate HMM models are not able to capture the interactions among different models. In many applications, multiple sequences are interacting with one another. For example, the actions of two arms of a human are obviously correlated/coupled to each other. In the sign language, the movements of different fingers on one hand can not be totally decoupled. Especially, in a game or battle, the action of one side will definitely affect or control the possible actions its opponent can take.

Recently, some extended or generalized HMM models have been used to solve various time-series and sequence data analysis problems, such as complex human action recognition[9], protein sequence modeling[12], traffic modeling[19] and biosignal analysis[23]. These new models usually aim to enrich the capabilities of standard HMM model by using more complex structures, while still being able to utilize the established methodologies (e.g. EM

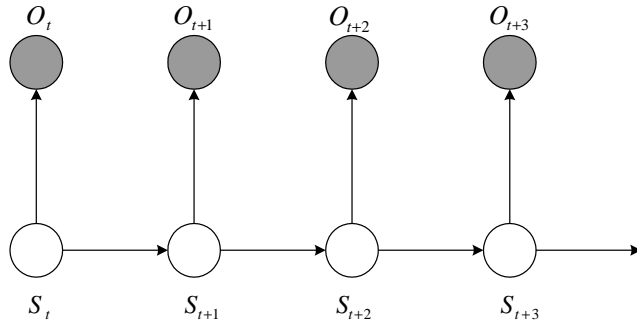


Figure 1: A first order HMM model. The empty circles are the hidden states and the shaded ones the observation nodes.

algorithm) for standard HMM models. Several typical examples from recent literature are coupled HMMs[8], event-coupled HMMs[18], factorial HMM[14] and input-output HMM[7]. More discussion and comparison of these model structures are presented in section 3.1.

The focus of this paper is on a new CHMM formulation that we propose to model the coupled relationships between multiple sequences. In this new formulation, not only do we want to couple multiple HMM models, we also intend to have some parameters to directly characterize the coupling relationships. To this end, we propose to simplify the joint conditional probability formulation. We model it as a linear combination of marginal conditional probabilities with the weights represented by coupling coefficients. Later in section 3, we describe the extended forward/backward procedures and learning algorithms for the new CHMM formulation. And it can be seen that these algorithms solve the problems for our new CHMM in an elegant way.

The advantages of our proposed architecture lie in several aspects. First, the capability of modeling multiple interacting sequences is added to standard HMM models. Second, we get reduced number of parameters compared to standard fully-coupled HMM as detailed in section 3.1. And most importantly, coupling coefficients are introduced to directly characterize the coupling. This is very useful in that we can get a measure of coupling between two objects. For example, if we know in some application the coupling strength is a function of distance, we can then get some idea about the distance by estimating the coupling coefficients through the coupled HMM learning algorithm described in this paper.

The organization of this paper is as follows. Section 2 reviews the standard HMM. Section 3 presents our new CHMM formulation and the corresponding extended forward-backward procedure and training algorithms. Section 4 discusses two experimental results, on artificial data and realistic data, respectively. Finally, section 5 concludes this paper.

2 Standard hidden Markov model

2.1 Elements of HMM

Very different notations have been used by different authors for describing the parameters of an HMM. Following convention used in [22], we describe the elements of a first order HMM model with discrete observations as follows:

- *set of hidden states* S_1, S_2, \dots, S_N , where N is the number of hidden states in the model;
- *initial state probability distribution* $\pi = \{\pi_i\}$, where for $1 \leq i \leq N$,

$$\pi_i = P(q_1 = S_i), \pi_i \geq 0, \text{ and } \sum_{i=1}^N \pi_i = 1 \quad (1)$$

- *state transition probability matrix* $A = \{a_{ij}\}$, where for $1 \leq i, j \leq N$,

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), a_{ij} \geq 0, \text{ and } \sum_{j=1}^N a_{ij} = 1 \quad (2)$$

where q_t is the hidden state at time t . For simplicity, sometimes in this paper we use $S_{j,t}$ to represent $q_t = S_j$, and thus $a_{ij} = P(S_{j,t+1} | S_{i,t})$.

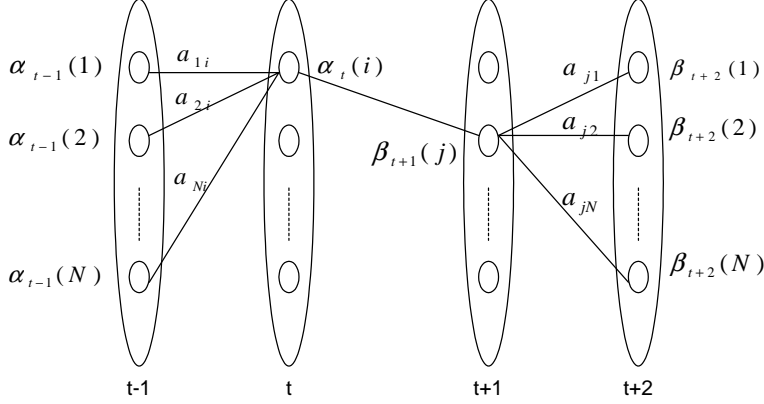


Figure 2: Illustration of forward-backward procedure. $\alpha_t(i)$ is the forward variable for state i at time t and $\beta_t(j)$ the backward variable for state j at time t .

- set of observation/output symbols $V = v_1, v_2, \dots, v_M$, where M is the number of observation symbols (i.e. the size of output alphabet);
- observation probability matrix $B = \{b_j(k)\}$, where for $1 \leq j \leq N$,

$$b_j(k) = P(o_t = v_k | q_t = S_j), \quad b_j(k) \geq 0, \quad \text{and} \quad \sum_{k=1}^M b_j(k) = 1 \quad (3)$$

So a standard HMM is usually denoted as a triplet $\lambda = (\pi, A, B)$.

2.2 Forward-backward procedure

The basic calculation of the observation evaluation problem $P(O|\lambda)$ is as follows

$$P(O|\lambda) = \sum_S P(O|S)P(S|\lambda) \quad (4)$$

where S is any possible hidden state sequence. Let the number of hidden states for model λ be N and the length of observation sequence T , we will have N^T possible hidden state sequences for model λ in total. This is obviously an intractable number. So in practice, more efficient algorithms such as forward and backward algorithms have been developed to tackle this problem.

Let $O = (o_1 o_2 \dots o_T)$ be an observation sequence where $o_t \in V$ is the observation symbol at time t . Given a model λ and an observation sequence O , $P(O|\lambda)$ can be solved using forward and/or backward variables (Fig. 2):

- forward variable $\alpha_t(i)$ is defined as the probability of the partial observation sequence $o_1 o_2 \dots o_t$ with state $q_t = S_i$ given model λ

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, S_{i,t} | \lambda) \quad (5)$$

which can be calculated inductively:

a) initialization

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \quad (6)$$

b) induction

$$\alpha_t(i) = b_i(o_{t+1}) \sum_{j=1}^N a_{ij} \alpha_{t-1}(j), \quad 2 \leq t \leq T, \quad 1 \leq i \leq N \quad (7)$$

- backward variable $\beta_t(i)$ is defined as the probability of the partial observation sequence $o_{t+1} o_{t+2} \dots o_T$ given state $q_t = S_i$ and model λ

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T | S_{i,t}, \lambda) \quad (8)$$

which can be calculated inductively:

a) initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad (9)$$

b) induction

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad 1 \leq t \leq T-1, \quad 1 \leq i \leq N \quad (10)$$

• *observation evaluation*

$$P(O|\lambda) = \sum_{i=1}^N \alpha_t(i) \beta_t(i), \quad \forall t \quad (11)$$

especially,

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (12)$$

It is easy to see that the computational complexity of the forward-backward procedure is $O(TN^2)$.

2.3 Baum-Welch training algorithm

Now let us consider the model training problem: Given an observation sequence O , how do we find the optimal model parameter vector λ that maximizes $P(O|\lambda)$. Given finite observations, there is no analytical or easy way of estimating the model parameters. Usually, an iterative procedure such as Baum-Welch method (or equivalently the EM algorithm[11]) is used to train the parameters such that the likelihood $P(O|\lambda)$ is locally maximized. Basically, the classical work of Baum and his colleagues[3, 4, 5] set the basis of a widely used training algorithm for HMM. Since the training algorithm used for our problem is based on their work too, we want to discuss the Baum-Welch algorithm in some depth here.

Baum *et al* found a group of reestimation formulas for the HMM parameters subject to stochastic constraints. These formulas can be explained either by intuitive bayesian posteriori reestimation or by standard optimization method. For the second interpretation, they constructed a self-mapping transformation based on the optimality equations from the Lagrange multiplier method and proved the transformation, when applied to HMM parameters, leads to an increase in the objective function $P(O|\lambda)$. They also constructed an auxiliary function to prove that and thus the convergence of the iterative algorithm based on the transformation. The auxiliary function and the derivation process described in [3] lead to an obvious EM interpretation of Baum-Welch algorithm and a solution to training our new CHMM, detailed later in this paper.

In Baum's work, the auxiliary function is defined as

$$Q(\lambda, \lambda') = \sum_S P(O, S|\lambda) \log P(O, S|\lambda') \quad (13)$$

where S is any particular state sequence and λ' is the auxiliary variable that corresponds to λ . And the following theorem was proved using Jensen's inequality.

Theorem 1 [3] *If $Q(\lambda, \lambda') \geq Q(\lambda, \lambda)$, then $P(O|\lambda') \geq P(O|\lambda)$ and the equality holds if and only if $\lambda' = \lambda$.*

Standard Lagrange mulitplier optimization method was applied to solving for λ' maximizing $Q(\lambda, \lambda')$ and the following parameter reestimation formulas based on forward and backward variables (Fig. 2) were generated:

a) state transition probability

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to } S_j}{\text{expected number of transitions from state } S_i} \quad (14a)$$

$$= \frac{\sum_t P(S_{i,t}, S_{j,t+1} | O, \lambda)}{\sum_t P(S_{i,t} | O, \lambda)} \quad (14b)$$

$$= \frac{\sum_t \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_t \alpha_t(i) \beta_t(i)} \quad (14c)$$

$$= \frac{a_{ij} (\partial P(O|\lambda) / \partial a_{ij})}{\sum_k a_{ik} (\partial P(O|\lambda) / \partial a_{ik})} \quad (14d)$$

b) observation probability

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j} \quad (15a)$$

$$= \frac{\sum_{t, O_t=v_k} P(S_{j,t}|O, \lambda)}{\sum_t P(S_{j,t}|O, \lambda)} \quad (15b)$$

$$= \frac{\sum_{t, O_t=v_k} \alpha_t(j)\beta_t(j)}{\sum_t \alpha_t(j)\beta_t(j)} \quad (15c)$$

$$= \frac{b_j(k)(\partial P(O|\lambda)/\partial b_j(k))}{\sum_l b_j(l)(\partial P(O|\lambda)/\partial b_j(l))} \quad (15d)$$

c) prior probability

$$\bar{\pi}_i = \text{expected number of times at state } i \text{ at time } t = 1 \quad (16a)$$

$$= P(S_{i,1}|O, \lambda) \quad (16b)$$

$$= \frac{\alpha_1(i)\beta_1(i)}{\sum_k \alpha_1(k)\beta_1(k)} \quad (16c)$$

$$= \frac{\pi_i(\partial P(O|\lambda)/\partial \pi_i)}{\sum_k \pi_k(\partial P(O|\lambda)/\partial \pi_k)} \quad (16d)$$

The interpretation of these reestimation equations is straightforward. \bar{a}_{ij} is just the ratio of the expected number of transitions from state S_i to S_j and the expected number of times being at state S_i . Similar statements can be made for $\bar{b}_j(k)$ and $\bar{\pi}_i$. By applying the above reestimation iteratively, eventually the likelihood function converges to a critical point. The solution provided by this algorithm is a so-called maximum likelihood (ML) estimation of HMM parameters.

Baum and his co-workers also found that these reestimation formulas can be written in the form of a self-mapping transformation as shown in Eq. (14d), (15d) and (16d). As mentioned above, these transformations can be explained as an attempt to solve the likelihood maximization problem using standard optimization method. By setting the first derivatives of Lagrangian

$$\mathcal{L} = P(O|\lambda) + \sum_i \mu_i \left(\sum_j a_{ij} - 1 \right) + \sum_j \nu_j \left(\sum_k b_j(k) - 1 \right) + \omega \left(\sum_i \pi_i - 1 \right) \quad (17)$$

to zeros, it can be shown that $P(O|\lambda)$ is maximized when

$$a_{ij} = \frac{a_{ij}(\partial P(O|\lambda)/\partial a_{ij})}{\sum_k a_{ik}(\partial P(O|\lambda)/\partial a_{ik})} \quad (18)$$

$$b_j(k) = \frac{b_j(k)(\partial P(O|\lambda)/\partial b_j(k))}{\sum_l b_j(l)(\partial P(O|\lambda)/\partial b_j(l))} \quad (19)$$

$$\pi_i = \frac{\pi_i(\partial P(O|\lambda)/\partial \pi_i)}{\sum_k \pi_k(\partial P(O|\lambda)/\partial \pi_k)} \quad (20)$$

Interestingly, the reestimation formulas (Eq. (14d), (15d) and (16d)) suggested by the above optimality equations are guaranteed to converge to a local maximum of $P(O|\lambda)$ by the following theorem.

Theorem 2 [4] *Let \mathcal{P} be a homogeneous polynomial*

$$\mathcal{P}(z_1, \dots, z_n) = \sum_{\mu_1, \mu_2, \dots, \mu_n} c_{\mu_1, \mu_2, \dots, \mu_n} z_1^{\mu_1} z_2^{\mu_2} \dots z_n^{\mu_n} \quad (21)$$

where $c_{\mu_1, \mu_2, \dots, \mu_n} \geq 0$ and $\mu_1 + \dots + \mu_n = d$. Then

$$\tau : z_i \rightarrow \frac{z_i \partial \mathcal{P} / \partial z_i}{\sum_j z_j \partial \mathcal{P} / \partial z_j} \quad (22)$$

maps $D : z_i \geq 0, \sum z_i = 1$ into itself and satisfies $\mathcal{P}(\tau(z_i)) \geq \mathcal{P}(z_i)$. In fact, strict inequality holds unless z_i is a critical point of \mathcal{P} in D .

This is a very interesting result. It basically says that we can forget about the auxiliary function and EM algorithm and use the iterative algorithm based on transformation τ to locally optimize the objective function P , if P is a polynomial of the above form and its parameters are subject to stochastic constraints. Later in section 3.5.2, we show that this interesting fact leads to a straightforward training algorithm for our CHMM formulation.

2.4 Continuous observation case

All our discussion, to this point, has considered only the case when the observations were characterized as discrete symbols from a finite alphabet. In a wide variety of applications, the observations are often continuous time-series signals (e.g. biosignal, speech signal, etc.) In order to use a continuous observation density, some restrictions have to be placed on the form of the model probability density function (pdf) to ensure that the parameters of the pdf can be reestimated in a consistent way.

The most general representation of the pdf, for which reestimation procedure has been formulated [16, 22], is a finite mixture of the form

$$b_j(o) = \sum_{k=1}^M c_{jk} \mathcal{N}[o, \mu_{jk}, U_{jk}] \quad (23)$$

where o is the observation vector being modeled, c_{jk} the mixture coefficient, μ_{jk} the mean of the m -th mixture, and U_{jk} the covariance matrix for the k -th mixture in state j and \mathcal{N} is any log-concave or elliptically symmetric density function. Usually a Gaussian density function is used for \mathcal{N} and the mixture gains(weights) c_{jk} satisfy the stochastic constraint

$$\sum_{k=1}^M c_{jk} = 1, \text{ and } c_{jk} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad (24)$$

so that the pdf is properly normalized, i.e.

$$\int_{-\infty}^{\infty} b_j(x) dx = 1, \quad 1 \leq j \leq N \quad (25)$$

The pdf of Eq. (23) can be used to approximate, arbitrarily closely, any finite, continuous density function. Hence it can be applied to a wide range of problems.

It can be shown that the reestimation equations for the coefficients of the mixture density, i.e. c_{jk} , μ_{jk} , and U_{jk} , are of the form

$$\bar{c}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{m=1}^M \sum_{t=1}^T \gamma_t(j, m)} \quad (26)$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot o_t}{\sum_{t=1}^T \gamma_t(j, k)} \quad (27)$$

$$\bar{U}_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot (o_t - \mu_{jk})(o_t - \mu_{jk})^T}{\sum_{t=1}^T \gamma_t(j, k)} \quad (28)$$

where $\gamma_t(j, k)$ is the probability of being in state j at time t with the k -th mixture component accounting for o_t , i.e.

$$\gamma_t(j, k) = \frac{\alpha_t(j) \beta_t(j)}{\sum_i \alpha_t(i) \beta_t(i)} \cdot \frac{c_{jk} \mathcal{N}(o_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^M c_{jm} \mathcal{N}(o_t, \mu_{jm}, U_{jm})} \quad (29)$$

The reestimation formula for a_{ij} is identical to the one used for discrete observation densities. The interpretation for Eq. (26)-(28) is fairly straightforward. One can easily generate arguments that are similar to those discussed in last section.

3 Coupled hidden Markov models

3.1 Various new HMM architectures

There are a variety of new HMM architectures that have been proposed to address a specific class of problems and to overcome certain limitations in the traditional HMM. Here we shall just discuss several of them (shown in Fig. 3) that are related to our proposed approach.

The standard fully-coupled HMMs (Fig. 3(a)) generally refer to a group of HMM models in which the state of one model at time t depends on the states of all models (including itself) at time $t - 1$. This is probably the

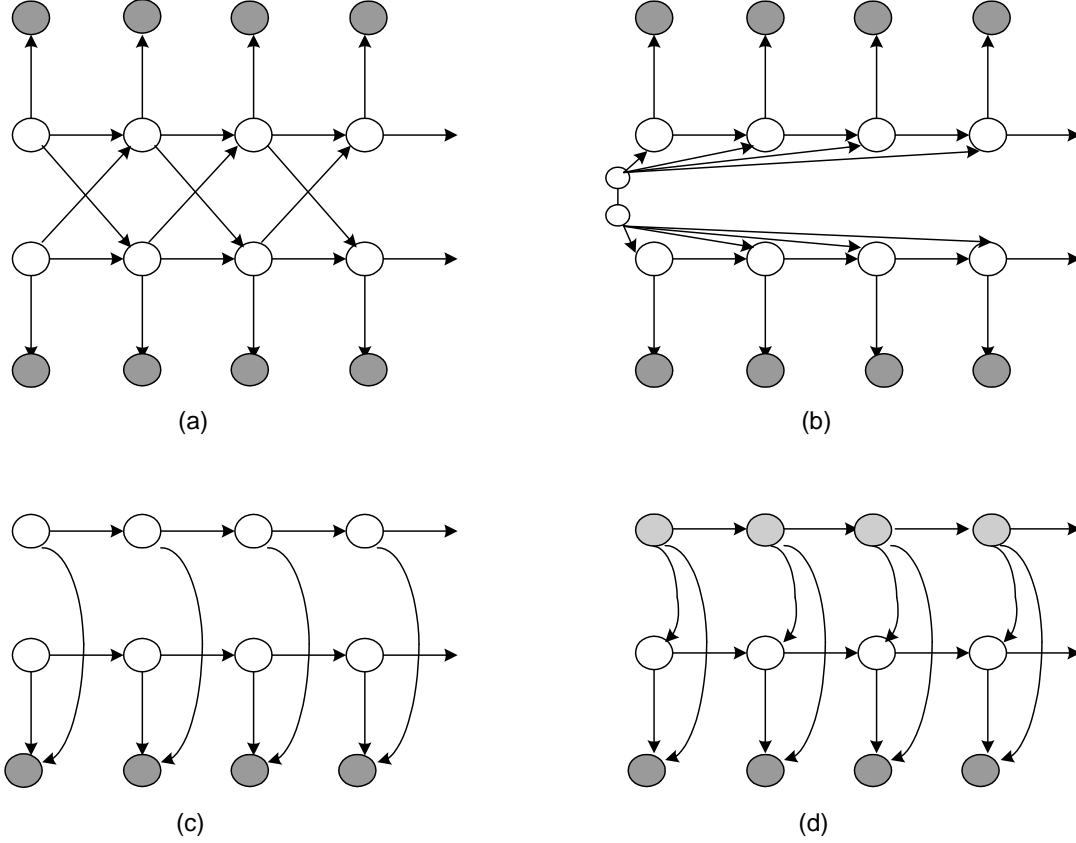


Figure 3: Various new HMM architectures. The empty circles are the hidden states and the shaded ones the observation nodes (except for (d) the light shaded ones are the input nodes). (a) Coupled HMMs; (b) Event-coupled HMMs; (c) Factorial HMMs; (d) Input-Output HMM.

most natural structure one can come up with. For C HMMs coupled together the state transition probability is usually described as $P(S_t^{(c)} | S_{t-1}^{(1)}, S_{t-1}^{(2)}, \dots, S_{t-1}^{(C)})$ instead of $P(S_t^{(c)} | S_{t-1}^{(c)})$ as in a single standard HMM model. In other words, the state transition probability is described by a $(C + 1)$ dimensional matrix and the number of free parameters for this transition probability matrix is N^C , which is exponential in the number of models coupled together (assume the number of hidden state N is the same for all models). Obviously, this is not a desirable feature because it makes accurate parameter learning very difficult.

There have been some variations of the fully-coupled HMMs and some interesting new HMM architectures for which the model size and inference problem are more tractable than the standard fully-coupled HMM models. Coupled HMMs[8] proposed by Matthew Brand is one of them. In his paper, Brand substituted the joint conditional dependency by the product of all marginal conditional probabilities, i.e.

$$P(S_t^{(c)} | S_{t-1}^{(1)}, S_{t-1}^{(2)}, \dots, S_{t-1}^{(C)}) = \prod_{c'=1}^C P(S_t^{(c)} | S_{t-1}^{(c')}) \quad (30)$$

This simplification reduced the transition probability parameter space. It has been used for recognizing complex human actions/behaviors[8, 9]. But Brand did not give any assumption or condition under which this equation can hold. Actually, one can see that if at each time slice the hidden states of different models are independent of one another, then the following equation holds.

$$P(S_t^{(c)} | S_{t-1}^{(1)}, S_{t-1}^{(2)}, \dots, S_{t-1}^{(C)}) = \frac{\prod_{c'=1}^C P(S_t^{(c)} | S_{t-1}^{(c')})}{P(S_t^{(c)})^{C-1}} \quad (31)$$

In his formulation the denominator was missing without any reason. And in Brand's paper, there was no parameter to directly capture the interaction. In next section, we propose an alternative and more reasonable formulation

which reduces the parameter space and offers some parameters that can be used to capture coupling strength between HMMs.

Kwon and Murphy[19] used the fully-coupled HMMs to model freeway traffic and approximated the E-step (of the EM algorithm) using particle filtering and Boyen-Koller algorithm. The approximation of E-step is similar to what we do for our new forward procedure and training algorithm. Our formulation has even lower complexity and the advantage of reduced parameter space.

Fig. 3(b) is a specific coupled HMMs described in [18] as *event-coupled HMMs*. The motivation there is to model a class of loosely coupled time series where only the onset of events are coupled in time. The representation power of event-coupled HMMs is obviously limited by the restrictive structure and this structure is for a very specific class of applications.

The factorial HMM[14] (Fig. 3(c)) enriches the representation power of hidden state by putting in multiple hidden state chains for one HMM. The model is still used to model one sequence, yet in a more complicated way. The hidden states may be set to be coupled but they are then coupled hidden states within one model. It does not intend to explore the interaction or dependency between two models.

IO-HMM[7] is addressing the input-output sequence pair modeling problem. While the IO-HMM may be viewed as a superset of coupled HMM (in which the hidden states from the previous time slice are treated as the inputs at current time slice), the input used in IO-HMM and hidden state from previous time slice are inherently different, certain independent assumption of inputs does not apply to the hidden states and the inference algorithm used in [7] is only for one HMM model and again not for general multiple coupled HMMs.

In some other papers, e.g. [23], simply use the joint probability, which is basically computationally intractable for even moderate number of models.

3.2 Proposed coupled HMMs

Obviously the fully coupled architecture (Fig. 3(a)) is the most powerful one to model interactions among multiple sequences. A lot of applications can be very naturally modeled by this structure, as discussed above. Our motivations are to leverage on this powerful architecture but introduce some parameters to directly capture the interaction and try to reduce the number of transition parameters in a reasonable way.

We propose to model the joint transition probability as

$$P(S_t^{(c)} | S_{t-1}^{(1)}, S_{t-1}^{(2)}, \dots, S_{t-1}^{(C)}) = \sum_{c'=1}^C \left(\theta_{c'c} P(S_t^{(c)} | S_{t-1}^{(c')}) \right) \quad (32)$$

where $\theta_{c'c}$ is the coupling weight from model c' to model c , i.e. how much $S_{t-1}^{(c')}$ affects the distribution of $S_t^{(c)}$. And how it affects is controlled by $P(S_t^{(c)} | S_{t-1}^{(c')})$. In other words, we model the joint dependency as a linear combination of all marginal dependencies. This formulation is flexible. For example, if we want to model the interaction among different objects as a function of distance, we can just make each $\theta_{c',c}$ a function of distance.

An interpretation for this formulation is presented below. For simplicity, we use y to represent the current state and x 's the previous states.

$$\begin{aligned} P(y|x_1, x_2, \dots, x_C) &= \frac{P(y, x_1, x_2, \dots, x_C)}{P(x_1, x_2, \dots, x_C)} \\ &= \frac{P(x_1)P(y|x_1)P(x_2|y, x_1) \cdots P(x_C|y, x_1, x_2, \dots, x_{C-1})}{P(x_1, x_2, \dots, x_C)} \\ &= w_1 P(y|x_1) \end{aligned} \quad (33)$$

where $w_1 = \frac{P(x_1)P(x_2|y, x_1) \cdots P(x_C|y, x_1, x_2, \dots, x_{C-1})}{P(x_1, x_2, \dots, x_C)}$. And similarly, we can have

$$P(y|x_1, x_2, \dots, x_C) = w_1 P(y|x_1) = w_2 P(y|x_2) = \cdots = w_C P(y|x_C) \quad (34)$$

with $\{w_c\}_{c=1, \dots, C}$ as appropriate coefficients that capture the complex dependencies between y and x 's. And finally we can rewrite the joint conditional probability as

$$P(y|x_1, x_2, \dots, x_C) = \sum_{c=1}^C \theta_c P(y|x_c) \quad (35)$$

where $\theta_c = \frac{1}{C} w_c$, $1 \leq c \leq C$, which are the parameters used to represent the interactions among different models in our formulation.

Our new formulation reduces the number of parameters compared to the standard CHMM. Of course the new model still contains more parameters than multiple standard HMMs. Precisely, we have C^2 transition probability matrices (compared to C in C standard HMMs) and one more coupling matrix Θ . Fortunately, in some practical applications, e.g. the speech recognition problem [22, 24], the number of output symbols is much larger than the number of hidden state so the increase in number of transition matrices does not increase the model complexity dramatically, provided C is not reasonably small (i.e. when there are not too much objects fully coupled together).

In short, the formulation we proposed is reasonable and easier than the standard fully-coupled HMMs to implement. And it has many potential applications. In next few sections the complete formulation is presented, including the forward-backward procedure and learning algorithms.

3.3 Extended parameter space in the coupled HMMs

Assume there are C coupled HMMs, the parameter space consists of the following components:

- prior probability $\pi = \{\pi_j^{(c)}\}$, $1 \leq c \leq C$, $1 \leq j \leq N^{(c)}$

$$\sum_{j=1}^{N^{(c)}} \pi_j^{(c)} = 1$$

- transition probability $A = \{a_{ij}^{(c',c)}\}$, $1 \leq c', c \leq C$, $1 \leq i \leq N^{(c')}$, $1 \leq j \leq N^{(c)}$

$$\sum_{j=1}^{N^{(c)}} a_{ij}^{(c',c)} = 1$$

- observation probability $B = \{b_j^{(c)}(k)\}$, $1 \leq c \leq C$, $1 \leq j \leq N^{(c)}$, $1 \leq k \leq M$

$$\sum_{k=1}^M b_j^{(c)}(k) = 1$$

- coupling coefficient $\Theta = \{\theta_{c'c}\}$, $1 \leq c', c \leq C$

$$\sum_{c'=1}^C \theta_{c'c} = 1$$

Thus, the proposed new coupled HMM models can be characterized by a quadruplet $\lambda = (\pi, A, B, \Theta)$, where Θ is the interaction parameters new in our formulation (as shown in Fig. 4). It can be seen that all these parameters are still subject to stochastic constraints, which provides conveniency for later reestimation procedure. All parameters can be estimated in a similar way (in discrete observation case).

3.4 Extended forward-backward procedure

For C coupled HMM's, similarly, we try to solve $P(O|\lambda)$ by defining the forward and backward variables. Note now each o_t is a vector $(o_t^{(1)} o_t^{(2)} \dots o_t^{(C)})^T$. Since the C HMMs are coupled together, the extended forward and backward variables should be defined jointly across C HMMs. In other words, we define the extended forward variable as

$$\alpha_t(j_1, j_2, \dots, j_C) = P(o_1, o_2, \dots, o_t, S_{t,j_1}, S_{t,j_2}, \dots, S_{t,j_C} | \lambda) \quad (36)$$

and the extended backward variable as

$$\beta_t(j_1, j_2, \dots, j_C) = P(o_{t+1}, \dots, o_T | S_{t,j_1}, S_{t,j_2}, \dots, S_{t,j_C}, \lambda) \quad (37)$$

Consequently, the inductive step for these two variables are

$$\alpha_t(j_1, j_2, \dots, j_C) = \begin{cases} \prod_c \pi_{j_c}^{(c)} \cdot b_{j_c}^{(c)}(o_1^{(c)}), & t = 1 \\ \sum_{i_1, i_2, \dots, i_C} \left(\alpha_{t-1}(i_1, i_2, \dots, i_C) \cdot \prod_c b_{j_c}^{(c)}(o_t^{(c)}) \sum_{c'} \theta_{c'c} \cdot a_{i_{c'} j_c}^{(c',c)} \right), & t > 1 \end{cases} \quad (38)$$

and

$$\beta_t(i_1, i_2, \dots, i_C) = \begin{cases} 1, & t = T \\ \sum_{j_1, j_2, \dots, j_C} \left(\beta_{t+1}(j_1, j_2, \dots, j_C) \cdot \prod_c b_{j_c}^{(c)}(o_{t+1}^{(c)}) \sum_{c'} \theta_{c'c} \cdot a_{i_{c'} j_c}^{(c',c)} \right), & t < T \end{cases} \quad (39)$$

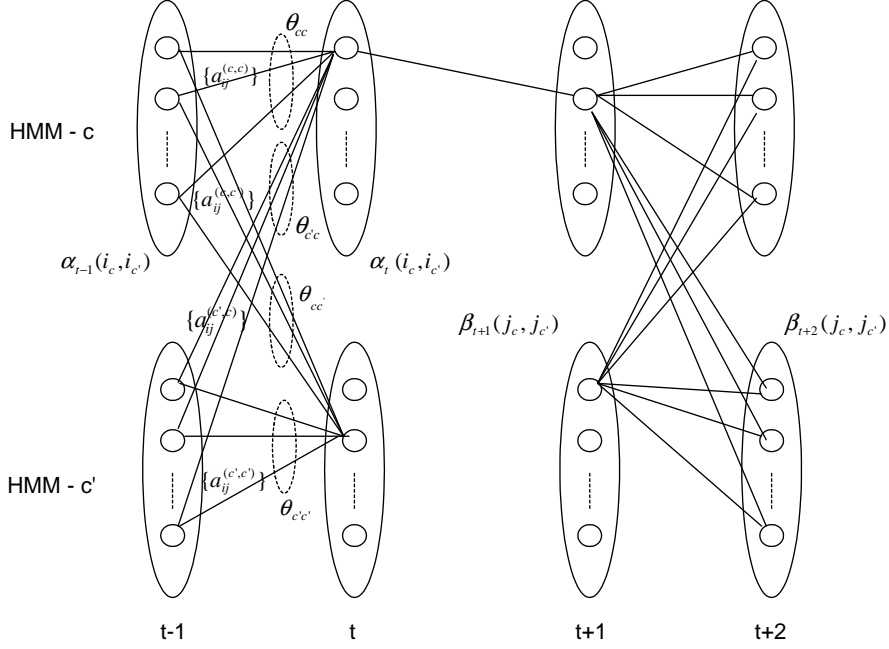


Figure 4: Extended forward-backward procedure. $\alpha_t(i_c, i_{c'})$ is the joint forward variable for CHMM (c, c') . $\beta_t(j_c, j_{c'})$ is the joint backward variable.

respectively. And the likelihood function $P(O|\lambda)$ can be solved as

$$P(O|\lambda) = \sum_{j_1, j_2, \dots, j_C} \alpha_T(j_1, j_2, \dots, j_C) \quad (40)$$

$$= \sum_{j_1, j_2, \dots, j_C} \alpha_t(j_1, j_2, \dots, j_C) \beta_t(j_1, j_2, \dots, j_C) \quad \forall t \quad (41)$$

It is easy to see that both the joint forward variable and backward variable can not be simply decoupled into a product of marginals, as mentioned in [23].

The computation is illustrated in Fig. 4. At each time slice there are N^C (if assume $N^{(c)} = N, \forall c$) possible β 's, which is an exponential number with respect to C . The computation complexity would be TN^C . So it is not practical to compute the forward-backward variables for moderate and large C . We propose to use a slightly modified forward variable which can be calculated for each HMM model separately and can reduce the computational complexity to $O(TCN^2)$. This modified forward variable is calculated inductively as follows:

a) Initialization:

$$\alpha_1^{(c)}(j) = \pi_j^{(c)} \cdot b_j^{(c)}(o_1^{(c)}), \quad 1 \leq j \leq N^{(c)} \quad (42)$$

b) Induction:

$$\alpha_t^{(c)}(j) = b_j^{(c)}(o_t) \sum_{c'=1}^C \theta_{c'c} \sum_{i=1}^{N^{(c')}} \left(\alpha_{t-1}^{(c')}(i) \cdot a_{ij}^{(c',c)} \right), \quad 2 \leq t \leq T \quad (43)$$

c) Termination:

$$P(O|\lambda) = \prod_{c=1}^C P^{(c)} = \prod_{c=1}^C \left(\sum_{j=1}^{N^{(c)}} \alpha_T^{(c)}(j) \right) \quad (44)$$

This may be seen as a mean-field approximation and the authors are currently looking at the possible interpretations. Experimental results show that $P(O|\lambda)$ calculated this way is close to true $P(O|\lambda)$. And the training algorithm based on this new forward variable produces reasonably good models.

3.5 Learning parameters for coupled HMMs

3.5.1 EM/GEM algorithm

EM and generalized EM (GEM) algorithm for estimating HMM parameters has been used in [7, 13]. The two basic steps in an EM or GEM algorithm, as described in [7], are

a) *estimation step*

Given observations O , parameters to estimate λ and the objective function $L(\lambda; O, S)$, an auxiliary function is constructed

$$Q(\lambda; \hat{\lambda}) = E_S[L(\lambda; O, S)|O, \hat{\lambda}] \quad (45)$$

which is the expectation of the objective over all possible state sequences, given observations O and the current estimate of the parameters $\hat{\lambda}$. Note $L(\cdot) = \log P(O, S|\lambda)$ in Baum-Welch algorithm.

b) *maximization step*

In exact EM algorithm this step is to solve the new estimated parameters by

$$\hat{\lambda}_{\text{new}} = \arg \max_{\lambda} Q(\lambda; \hat{\lambda}) \quad (46)$$

But often there is no way or it is too difficult to solve for the λ that maximizes $Q(\lambda, \hat{\lambda})$. In that case, we seek for a self-mapping transformation τ defined as $\hat{\lambda}_{\text{new}} = \tau(\hat{\lambda})$ such that

$$Q(\tau(\hat{\lambda}); \hat{\lambda}) \geq Q(\lambda; \hat{\lambda}) \quad (47)$$

This is the so-called generalized EM (GEM) algorithm. Obviously, the convergence of GEM algorithm is going to be slower than the original EM algorithm because in the "maximization" step the new estimate is selected just to increase $Q(\lambda; \hat{\lambda})$ instead of maximizing it.

Obviously, the Baum-Welch algorithm described in section 2.3 can be explained as an EM algorithm: Plug $P(O, S|\lambda) = P(O|\lambda)P(O|S, \lambda)$ into Eq. (13), we get

$$Q(\lambda, \lambda') = P(O|\lambda) \sum_S P(S|O, \lambda) \log P(O, S|\lambda') \quad (48)$$

If we view λ as the current estimate of the parameters and $Q(\lambda, \lambda')$ as a function of λ' , $P(O|\lambda)$ becomes just a scaling factor. Ignoring this scaling factor, $Q(\lambda, \lambda')$ can be viewed as the expectation of the log-likelihood function $\log P(O, S|\lambda')$ over the distribution of state sequences (S 's) given the observations O and the model λ . the reestimation, λ' , is set to be the solution to the maximization problem $\max_{\lambda'} Q(\lambda, \lambda')$.

In either Baum-Welch algorithm or the EM/GEM algorithm described for IO-HMM[7], the success of EM algorithm depends critically on the product form of $P(O, S|\lambda)$. In standard HMM and IO-HMM, with certain independence assumptions, $P(O, S|\lambda)$ (or $P(O, S|U, \lambda)$ for IO-HMM where U is the input sequence) can be written as a product of all kinds of probabilities along the state sequence S . After taking logarithm of this product, the summation (Eq. (48)) over all possible state sequences (corresponding to the expectation step of EM algorithm) can be easily computed in terms of forward and backward variables and the optimization with respect to each type of parameters (corresponding to the maximization step of EM algorithm) can be done separately. Due to the introduction of the coupling term, unfortunately, the quantity $P(S|O, \lambda)$ in our coupled HMM formulation is not of pure product form any more. The immediate consequence is that both the expectation and the maximization step become very difficult to compute for A and Θ since they are entangled together. Even though we can still generate reestimation equations similar to Eq. (15a) and (16a) for π and B , we don't have an efficient way of calculating the exact forward/backward variable any more so the exact EM solution becomes computationally very expensive.

The probability meaning of those transition matrices has been obscured by the introduction of coupling decomposition (i.e. modeling of the joint dependency using Eq. (32)). We have lost intuitive Bayesian interpretation for estimating any transition probability $P(S_{j,t+1}^{(c)}|S_{i,t}^{(c')})$ as shown in Fig. 4. So Eq. (14a) does not hold any more in our coupled HMM formulation.

3.5.2 Classical optimization technique

Using a little caution, one would find that Baum and his co-workers have contributed more than just the auxiliary function and EM algorithm for HMM. As described in section 2.3, they also described the self-mapping transformation τ (corresponding to Eq. (14d), (15d) and (15d)), which is motivated by the optimality condition of

standard Lagrange multiplier optimization method and leads to an iterative reestimation procedure. Even without the auxiliary function and EM algorithm, the Theorem 2 guarantees the convergence of the iterative algorithm based on the transformation. Levinson *et al* discussed this in [20] and presented a geometric interpretation of the transformation.

In this section, we apply the same techniques to derive an iterative optimization procedure for learning the parameters of our CHMM formulation. Following the derivation in [20], we work out the solution by constrained optimization techniques. The constraints are, of course, required to make the hidden Markov model well defined. It is thus natural to look at the training problem as a problem of constrained optimization of P and solve it by the classical method of Lagrange multipliers. For simplicity, we shall restrict the discussion to optimization with respect to A . Actually all parameters in $\lambda = (\pi, A, B, \Theta)$ are subject to similar stochastic constraints so the discussion with respect to A here can be easily duplicated for π , B and Θ . Let \mathcal{L} be the Lagrangian of P with respect to the constraints associated with A . We see that

$$\mathcal{L} = P + \sum_{i,c',c} \lambda_i^{(c',c)} \left(\sum_{j=1}^N a_{ij}^{(c',c)} - 1 \right) \quad (49)$$

where the $\lambda_i^{(c',c)}$ are the undetermined Lagrange multipliers. It is easy to verify that P is locally maximized when

$$a_{ij}^{(c',c)} = \frac{a_{ij}^{(c',c)} \partial P / \partial a_{ij}^{(c',c)}}{\sum_{k=1}^{N^{(c)}} a_{ik}^{(c',c)} \partial P / \partial a_{ik}^{(c',c)}} \quad (50)$$

Similar arguments can be made for π , B and Θ parameters. The reestimation formula suggested by the above equation is exactly the transformation (Eq. (22)) that has been discussed in [3, 20]. In the standard HMM case, with some manipulation on the transformation, we can get exactly the standard EM reestimation formulas.

While the likelihood function $P(O|\lambda)$ is more complicated in our coupled HMM formulation than in standard HMM case, it is easy to verify that P is still a homogeneous polynomial with respect to each type of parameters (namely, π , A , B and Θ) and each type of the parameters is still subject to stochastic constraints. So the elegant iterative solution to maximizing $P(O|\lambda)$ still applies in our couple HMMs case. Its convergence, again, is guaranteed by Theorem 2. Actually, the transformation τ can apply to more general likelihood functions which are polynomials with positive coefficients (not necessarily homogeneous). The relaxation was presented by Baum and Sell [5].

To summarize, our main points for this section are:

- a) In our coupled HMM case, we are not sure whether the transformation τ maximize the auxiliary function $Q(\cdot)$ or not. So we don't know if it still have an EM/GEM interpretation. And directly applying EM/GEM algorithm to learning our coupled HMM parameters is fairly difficult due to the linear combination introduced to model the joint conditional dependency.
- b) However, we can still easily derive the transformation from standard Lagrange multiplier optimization method because all parameters are still subject to stochastic constraints. And then based on the theorem proved by Baum and Eagon[4], it is easy to prove that by applying this transformation the reestimation of parameters will converge to a critical point of the likelihood function $P(O|\lambda)$.
- c) The reestimation formula is still elegant in that at each iteration, the stochastic constraints are kept automatically and it is guaranteed to converge to a local maxima. The only difference from the standard HMM case is that now we don't have a simple form of calculating the first derivative of the likelihood function. While in the standard HMM training algorithm, these derivatives reduce to a form in which only the forward and backward variables are needed. In our case, we have to calculate the derivatives using backpropagation through time. Fortunately, this does not add too much computational complexity because we can apply similar forward procedures for the calculation of derivatives and only one pass through time is needed to calculate the forward variable and all the derivatives. The detailed computation of these first derivatives are presented in Appendix A.

3.5.3 Other training algorithms

There are some other possible training algorithms that have been proposed for standard HMM model and we could consider in our future work. Here we mainly discuss two of them, gradient ascent method and genetic programming. These techniques may be (much) slower than the iterative algorithm present in previous section. But they feature

in different aspects: gradient-based approach is more appropriate for online learning and genetic approach excels in being able to finding global or better local optimum solutions.

Gradient-based algorithms for training HMM parameters have been examined by several authors [1, 2, 20, 25]. As discussed in [20], if the objective function is not $P(O|\lambda)$ or not of any special form, or the constraints of parameters change, the special transformation τ discussed above will not apply any more. That is exactly what happened in [1], in which they used gradient-based approach to train the parameters after changing the objective function from $P(O|\lambda)$ to something that reflects the mutual information of multiple separate HMM models. An advantage of this approach is that it is an online approach [2], which is suitable to incremental learning and situations where only limited computation resources are available.

A useful trick to satisfy the constraints at each iteration in gradient-based method is to fix the representation of any parameter x_j , with the constraint $\sum_j x_j = 1$ and $x_j \leq 0$, as [2]

$$x_j = \frac{e^{w_j}}{\sum_{k=1}^N e^{w_k}} \quad (51)$$

So the constraints are automatically satisfied at each iteration. The parameters x_j 's can be updated as

$$\Delta w_j^{(k+1)} = \zeta \Delta w_j^{(k)} + (1 - \zeta) \eta \frac{\partial P}{\partial w_j} \quad (52)$$

where η is the learning rate to be chosen by user and experimentation and ζ is the momentum rate used to accelerate the convergence. The partial derivatives $\partial P / \partial w_j$ can be calculated using back-propagation through time.

In order to be able to search for global optimum solution in the large and complex parameter space, genetic programming approach has been attempted to find the HMM parameters [10, 24]. It sounds exciting about global optimization. Even though there is no proof that genetic algorithm can find the global optimum eventually, it works pretty well in practice and has intuitive interpretations. The biggest disadvantage of this approach is that the computation time required is often prohibitive in practice.

3.5.4 Continuous observation case

As mentioned in previous sections, in many applications we need to model continuous (real-valued) observations. In these situations, we usually model the output as a mixture of gaussians (as discussed in section 2.4).

In our formulation, the forward variable is approximated and it is not likely to construct an approximated backward variable such that Eq. (11) is satisfied. There are several choices on how to train the parameters (c_{jk} 's, μ_{jk} 's and U_{jk} 's) associated with continuous observation. One simple choice is to approximate the real $\gamma_t(j, k)$ as

$$\hat{\gamma}_t(j, k) = \frac{\alpha_t(j)}{\sum_i \alpha_t(i)} \cdot \frac{c_{jk} \mathcal{N}(o_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^M c_{jm} \mathcal{N}(o_t, \mu_{jm}, U_{jm})} \quad (53)$$

and use the same reestimation formula as for standard HMM (Eq. (26), (27) and (28)). A second choice is to use an approximation for the backward variable and use the same reestimation formula just mentioned. For a third choice, we could use the transformation τ discussed in section 3.5.2 for $c_{jk}^{(c)}$ since the stochastic constraint holds. For $\mu_k^{(c)}$ and $\sigma_k^{(c)}$, the methods from either choice one or two can be used. In this paper, only results with choice one are presented since the authors are currently working on the other two alternatives.

3.5.5 Multiple observation samples

There have been some treatments [20, 21] on training HMM parameters with multiple observation samples. Usually by assuming the independency among the multiple samples, we have

$$P(O|\lambda) = \prod_{k=1}^K P(O_{(k)}|\lambda) \quad (54)$$

where each $O_{(k)}$ is an independent observation sample sequence. So the only change to our training algorithm (presented in section 3.5.2) is that the first order derivative needs to be calculated by summing over all samples, i.e.

$$\frac{\partial P(O|\lambda)}{\partial w} = \sum_k \frac{P(O|\lambda)}{P(O_{(k)}|\lambda)} \cdot \frac{\partial P(O_{(k)}|\lambda)}{\partial w} \quad (55)$$

4 Experimental results

4.1 Experimental setup

For artificial data, we sample from 2 coupled HMM models with 3 hidden states in each model/chain. And the number of observation states is 3 for discrete observations and 2 for continuous observations. As mentioned in [17], there are singularity problems with continuous formulation. To alleviate the problem, we set the number of observation states to 2 in continuous case. Transition probabilities A , observation probabilities B and the priors π are generated randomly. But the generating process is not totally random since the parameters are generated in such a way that the observation probability distribution for each hidden state is made very different. Because we found that if these parameters are initialized entirely at random the observation probability distribution for each hidden state is about the same or very close to each other. In that case, the difference between results of CHMM and non-coupled HMMs can barely be seen. Also it is trivial since different hidden states can then be merged into one. The other parameter Θ is set to be $\begin{bmatrix} 1-\theta & \theta \\ \theta & 1-\theta \end{bmatrix}$, where θ can be viewed as the coupling strength between two objects. We set it like this to explore how the performance of coupled HMMs vary with coupling strength.

For each constructed HMM model, 110 sample sequences of length $T=100$ are generated. 10 of them are used for training and 100 for testing. For training we assume we know the correct architecture, i.e. use the same number of hidden states and observation states as in the generating model. For training performance, we look at the log-likelihood of fitting training data to the trained model. The higher the likelihood, the better local maximum we think the training has converged to. The training performance is used to demonstrate how good our approximated formulation is. Recognition accuracy has been used to measure the testing performance.

The real data used for our experiment is downloaded from the UCI KDD Archive web site (<http://kdd.ics.uci.edu/>). It is an electroencephalography (EEG) time series data set. This data arises from a large study to examine EEG correlates of genetic predisposition to alcoholism. It contains measurements from 64 electrodes placed on the scalp sampled at 256 Hz (3.9-msec epoch) for 1 second. There are two types of objects - control objects and alcoholic objects. This is a fairly large data set. We just extracted a small fraction of the data that contains 10 measurements from 2 electrodes for each types of objects. This extraction is based on our limited understanding of the paper by L. Ingber[15]. So there are two feature sequences (from two electrodes) for each data sample.

4.2 Results and discussions on artificial data

The likelihood curves for discrete observations are shown in Fig. 5-6. Both the true and approximated likelihood values (actually log-likelihood) are shown for comparison. By true likelihood function, we refer to the exact inference for our CHMM formulation. Model training is based on our approximated likelihood function for Fig. 5 and on true likelihood function for Fig. 6 (starting from the same initial parameter values). It can be seen that optimizing the true likelihood function also increases the approximated likelihood. And the reverse argument generally holds as a global trend but not monotonically. More importantly, note that we can still reach a convergence point (local maximum) when optimizing over our approximated likelihood function. But it would usually be a worse local maximum than that can be reached by optimizing the true likelihood function. Fig. 7 and 8 show the same comparisons for continuous observations. We can get similar conclusions.

The recognition task in this experiment is to assign each test sequence to the correct model from which it is generated. For standard HMM models, we train two of them, one for each object. We then fit each test sequence to the two HMM models and assign the sequence to the one that has higher likelihood value. The accuracy for the standard HMM models measures how many test sequences have been assigned correctly (in percentage). We also train two CHMM models, one is using approximated inference, the other is the exact inference, both on our CHMM formulation. Assume we get two sequences at any time from two objects. The two sequences are presented to a CHMM model in all possible permutations, i.e. (sequence 1, sequence 2) and (sequence 2 sequence 1). The accuracy for CHMM model is the number of times that the model picks out the right permutation.

The recognition accuracy results on artificial data are shown in Fig. 9-10. We also plot the accuracy results for different θ 's and thus can see how recognition accuracy varies with the coupling strength. For discrete observations, our CHMM approach perform better than non-coupled HMMs and exact inference better than the approximated inference. Given that the exact inference is computationally impractical for more HMMs coupled together, the approximated inference of our CHMM formulation can be a good alternative in those situations. Of course, with more HMMs coupled together, the number of parameters for a CHMM increases and the quality of trained models may decrease. More investigations are needed to address this problem. For continuous observations, the results are not very clean. For θ smaller than 0.3, we observe the same trend as in discrete situation. But for larger θ ,

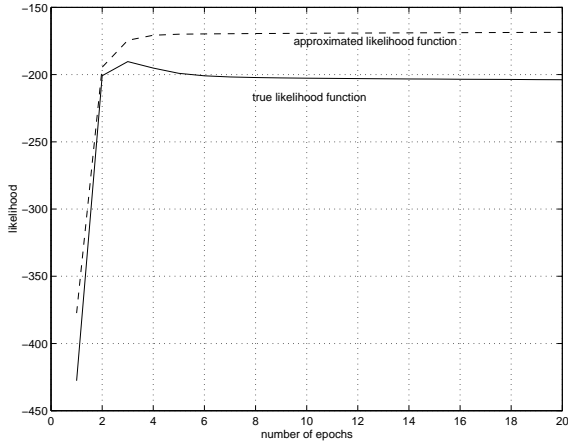


Figure 5: Comparison of likelihood functions when trained with approximated inference (with discrete observations)

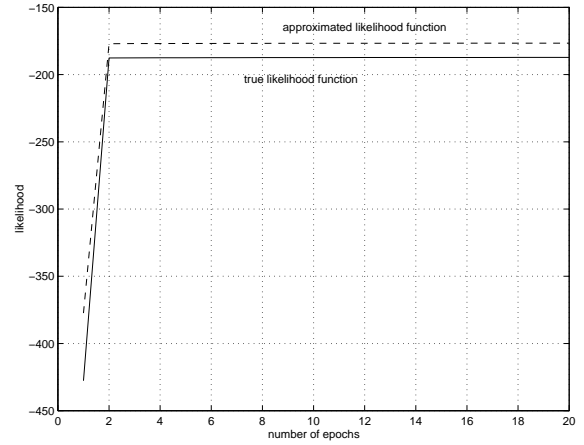


Figure 6: Comparison of likelihood functions when trained with exact inference (with discrete observations)

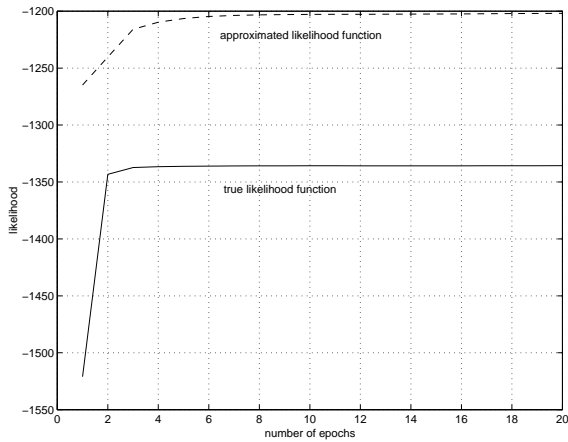


Figure 7: Comparison of likelihood functions when trained with approximated inference (with continuous observations)

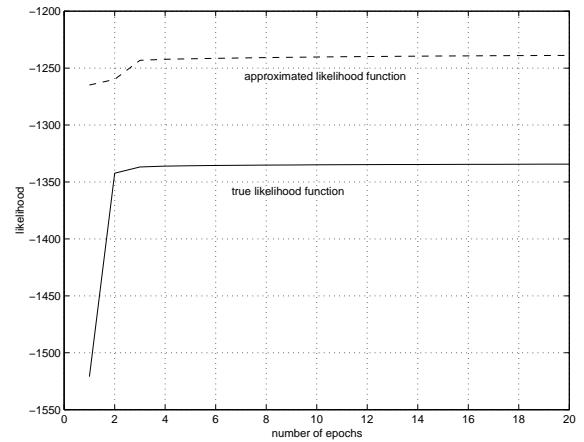


Figure 8: Comparison of likelihood functions when trained with exact inference (with continuous observations)

approximated inference of our CHMM formulation is worse than the exact inference and not better than standard non-coupled HMM models. The possible reason is that the approximation for continuous observation situation is getting worse as coupling becomes stronger so the CHMM model learned is not good. This encourages us to look at other alternative approximations described in section 3.5.4.

4.3 Results and discussions on real data

For each type of objects two standard HMM models and two CHMM models are trained. Each HMM model corresponding to one electrode sequence (feature). So we have four models to compare: HMM-1 - HMM model trained on feature 1; HMM-2 - HMM model trained on feature 2; CHMM-1 - our new CHMM formulation using approximated inference and two chains, one chain for each feature; CHMM-2 - our CHMM formulation using exact inference and two chains, one chain for each feature. Five-fold cross validation is used and we repeat the experiment for ten times to get the average accuracy (thus reducing the parameter initialization effect).

The recognition accuracy results for the extracted EEG data are shown in Table 1. Our CHMM approach does produce better results than standard HMM models. This may be natural since the CHMM approach uses two features together and single HMM model is trained on one of the two features. But since there is no easy way of combining two single HMM's results, CHMM approach has a natural advantage of using two feature sequences simultaneously. The exact inference delivers better accuracy than approximated inference. This is not surprising

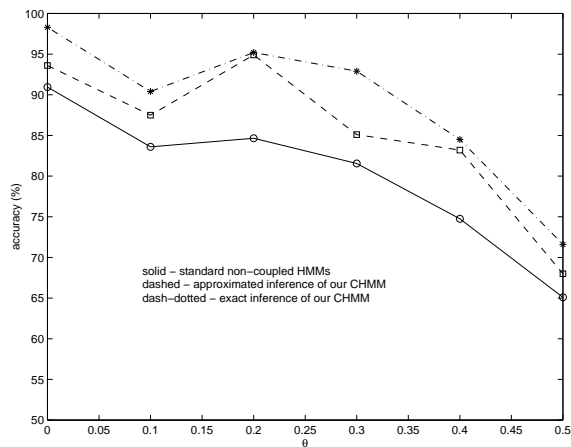


Figure 9: Recognition accuracy (with discrete observations)

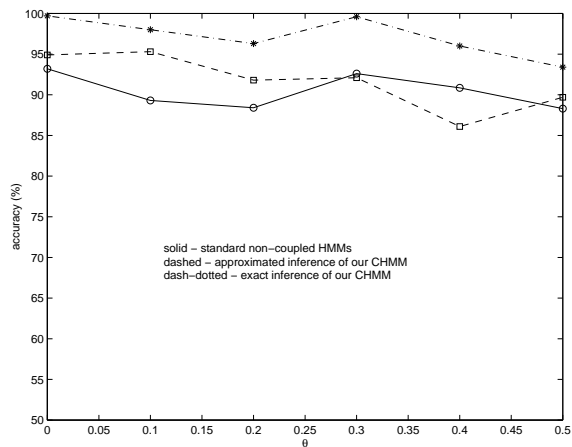


Figure 10: Recognition accuracy (with continuous observations)

but we believe the our approximation used in the formulation is still coarse, especially for continuous observations. And we expect to get better results when using alternatives described in section 3.5.4. We are currently working in that direction.

Table 1: Recognition accuracy results for EEG data

Model	Accuracy (%)
HMM-1	69
HMM-2	54
CHMM-1	75.5
CHMM-2	81.5

5 Conclusion and future work

In this paper we present a new formulation of CHMM that has reduced parameter space compared to standard CHMM and performs better for the recognition tasks in our experiments. Approximated forward/backward procedure and training algorithm for our new CHMM formulation are proposed to make the computation practical. The results are, however, not very satisfying in that the classification/recognition results using approximated inference are not very close to exact inference. As discussed in previous sections, we expect to get better results with alternative approximations for continuous observation case.

Future work can proceed in several directions:

- *Improved objective functions.* For example, we could use some mutual information objective function similar to the one used in [1]. Such mutual information objective function should be natural and appropriate for CHMM approach since here we are dealing with multiple sequences simultaneously. And using a different objective function couple result in improved quality of trained models.
- *Analysis of the size of data/model versus the accuracy of learning couple HMM parameters.* Our CHMM model has the advantage of computation efficiency and smaller parameter space compared to standard CHMM. But as the size (complexity) of the model grows, the quality of model learned from limited data may decrease. Experimental analysis on this issue is one of our future work.
- *More applications.* There are potentially many more applications to which our CHMM approach can be applied. For example, in Sign language recognition, five fingers are coupled and each of them may be modeled by one HMM. The main challenge for real data is that appropriate preprocessing of the data is often critical to the success of modeling task. Another important aspect is to see if we can get meaningful interpretation the interaction parameter θ in real applications.

Appendix

A Partial derivatives

The partial derivatives used in our parameter learning algorithm for coupled HMMs can be calculated as follows: For simplification purpose, we let $\delta_{x,y} = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}$ and $z_{ijt}^{(c',c)} = \theta_{c'c} \cdot a_{ij}^{(c',c)} \cdot b_j^{(c)}(o_t^{(c)})$. Let $\vec{w} = (\pi, A, B, \Theta)$ be the parameter vector, we have

$$\alpha_t^{(c)}(j) = \begin{cases} \pi_j^{(c)} \cdot b_j^{(c)}(o_1^{(c)}), & t = 1 \\ \sum_{c'} \sum_i z_{ijt}^{(c',c)} \alpha_{t-1}^{(c')}(i), & 2 \leq t \leq T \end{cases} \quad (\text{A-1})$$

$$\frac{\partial P}{\partial w} = \sum_c \left(\frac{P}{P^{(c)}} \frac{\partial P^{(c)}}{\partial w} \right) = \sum_c \left(\frac{P}{P^{(c)}} \sum_{j=1}^N \frac{\partial \alpha_T^{(c)}(j)}{\partial w} \right) \quad (\text{A-2})$$

By using the fact

$$\frac{\partial \alpha_t^{(c)}(j)}{\partial w} = \sum_{c'} \sum_i \left(\frac{\partial z_{ijt}^{(c',c)}}{\partial w} \alpha_{t-1}^{(c')}(i) + z_{ijt}^{(c',c)} \frac{\partial \alpha_{t-1}^{(c')}(i)}{\partial w} \right), \quad 2 \leq t \leq T \quad (\text{A-3})$$

we get the first order derivatives of $\alpha_t^{(c)}(j)$ with respect to each type of parameters as follows.

- $\partial \alpha_t^{(c)}(j) / \partial \pi_i^{(c_1)}$

$$\frac{\partial \alpha_t^{(c)}(j)}{\partial \pi_i^{(c_1)}} = \begin{cases} \delta_{ij} \delta_{c,c_1} \cdot b_j^{(c_1)}(o_1^{(c_1)}), & t = 1 \\ \sum_{c'=1}^C \sum_{k=1}^{N^{(c')}} z_{kjt}^{(c',c)} \frac{\partial \alpha_{t-1}^{(c')}(k)}{\partial \pi_i^{(c_1)}}, & 2 \leq t \leq T \end{cases} \quad (\text{A-4})$$

- $\partial \alpha_t^{(c)}(j) / \partial a_{ij}^{(c',c)}$

$$\frac{\partial \alpha_t^{(c)}(j)}{\partial a_{i_1 j_1}^{(c_1, c_2)}} = \begin{cases} 0, & t = 1 \\ \delta_{c,c_2} \delta_{j,j_1} \theta_{c_1 c_2} b_{j_1}^{(c_2)}(o_t^{(c_2)}) \alpha_{t-1}^{(c_2)}(i_1) + \sum_{c'} \sum_i z_{ijt}^{(c',c)} \frac{\partial \alpha_{t-1}^{(c')}(i)}{\partial a_{i_1 j_1}^{(c_1, c_2)}}, & 2 \leq t \leq T \end{cases} \quad (\text{A-5})$$

- $\partial \alpha_t^{(c)}(j) / \partial b_{j_1}^{(c_1)}(k)$

$$\frac{\partial \alpha_t^{(c)}(j)}{\partial b_{j_1}^{(c_1)}(k)} = \begin{cases} \delta_{o_1^{(c)}, k} \delta_{c,c_1} \delta_{j,j_1} \pi_{j_1}^{c_1}, & t = 1 \\ \sum_{c'} \sum_i \left(\delta_{o_1^{(c)}, k} \delta_{c,c_1} \delta_{j,j_1} \theta_{c' c_1} a_{ij_1}^{(c',c_1)} \alpha_{t-1}^{(c')}(i) + z_{ijt}^{(c',c)} \frac{\partial \alpha_{t-1}^{(c')}(i)}{\partial b_{j_1}^{(c_1)}(k)} \right), & 2 \leq t \leq T \end{cases} \quad (\text{A-6})$$

- $\partial \alpha_t^{(c)}(j) / \partial \theta_{c_1 c_2}$

$$\frac{\partial \alpha_t^{(c)}(j)}{\partial \theta_{c_1 c_2}} = \begin{cases} 0, & t = 1 \\ \delta_{c,c_2} \sum_i a_{ij}^{(c_1, c_2)} b_j^{(c_2)}(k) \alpha_{t-1}^{(c_1)}(i) + \sum_{c'} \sum_i z_{ijt}^{(c',c)} \frac{\partial \alpha_{t-1}^{(c')}(i)}{\partial \theta_{c_1, c_2}}, & 2 \leq t \leq T \end{cases} \quad (\text{A-7})$$

In continuous observation case, we model the output as

$$b_j^{(c)}(o_t^{(c)}) = \sum_{k=1}^M c_{jk}^{(c)} \mathcal{N}(o_t^{(c)}, \mu_k^{(c)}, \sigma_k^{(c)}) \quad (\text{A-8})$$

The partial derivative of $P(O|\lambda)$ with respect to the $c_{jk}^{(c)}$ is

$$\frac{\partial P}{\partial c_{jk}^{(c)}} = \frac{\partial P}{\partial b_j^{(c)}(o_t^{(c)})} \cdot \frac{\partial b_j^{(c)}(o_t^{(c)})}{\partial c_{jk}^{(c)}} \quad (\text{A-9})$$

and we can calculate the two product elements separately. From the derivations above, we can clearly see that the first one amounts to calculating $\frac{\partial \alpha_t^{(c)}(j)}{\partial b_{j_1}^{(c_1)}(o_t^{(c_1)})}$ as follows:

$$\frac{\partial \alpha_t^{(c)}(j)}{\partial b_{j_1}^{(c_1)}(o_t^{(c_1)})} = \begin{cases} \delta_{c,c_1} \delta_{j,j_1} \pi_{j_1}^{(c_1)}, & t = 1 \\ \sum_{c'} \sum_i \left(\delta_{c,c_1} \delta_{j,j_1} \theta_{c'c_1} a_{ij_1}^{(c',c_1)} + z_{ij}^{(c',c)} \frac{\partial \alpha_{t-1}^{(c')}(i)}{\partial b_{j_1}^{(c_1)}(o_t^{(c_1)})} \right), & 2 \leq t \leq T \end{cases} \quad (\text{A-10})$$

And the second one is simply

$$\frac{\partial b_j^{(c)}(o_t^{(c)})}{\partial c_{jk}^{(c)}} = \mathcal{N}(o_t^{(c)}, \mu_k^{(c)}, \sigma_k^{(c)}) \quad (\text{A-11})$$

References

- [1] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 49–52, Tokyo, Japan, April 1986.
- [2] P. Baldi and Y. Chauvim. Smooth on-line algorithms for hidden Markov models. *Neural Computation*, 6:307–318, 1994.
- [3] L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8, 1969.
- [4] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. *Bulletin AMS*, 73:360–363, 1967.
- [5] L. E. Baum and G. R. Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, pages 211–227, 1968.
- [6] Y. Bengio. Markovian models for sequential data. *Neural Computing Surveys*, 2:129–162, 1999.
- [7] Y. Bengio and P. Frasconi. Input-Output HMMs for sequence processing. *IEEE Trans. Neural Networks*, 7(5):1231–1249, September 1996.
- [8] Matthew Brand. Coupled hidden Markov models for modeling interactive processes. Technical Report 405, MIT Media Lab, 1997.
- [9] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 994–999, 1997.
- [10] C. W. Chau, S. Kwong, C. K. Diu, and W. R. Fahrner. Optimization of HMM by a genetic algorithm. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 3, pages 1727–1730, 1997.
- [11] A. P. Dempster, N. M. Laird, , and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.
- [12] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14:755–763, 1998.
- [13] Y. Ephraim, A. Dembo, and L. R. Rabiner. A minimum discrimination information approach for hidden Markov modeling. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Dallas, TX, April 1987.
- [14] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997.
- [15] L. Ingber. Statistical mechanics of neocortical interactions: Canonical momenta indicators of electroencephalography. *Physical Review E*, 55(4):4578–4593, 1997.
- [16] B.-H. Juang. Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, 64(6):1235–1249, 1985.
- [17] B.-H. Juang, S. E. Levinson, and M. M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of Markov chains. *IEEE Trans. Inform. Theory*, 32(2):307–309, 1986.
- [18] T. T. Kristjansson, B. J. Frey, and T. Huang. Event-coupled hidden Markov models. In *Proc. IEEE Int. Conf. on Multimedia and Exposition*, volume 1, pages 385–388, 2000.
- [19] J. Kwon and K. Murphy. Modeling freeway traffic with coupled HMMs. Technical report, University of California at Berkeley, May 2000.
- [20] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell System Technical Journal*, 62(7):1035–1074, April 1983.
- [21] X. Li, M. Parizeau, and R. Plamondon. Training hidden Markov models with multiple observations - a combinatorial method. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(4):371–377, April 2000.

- [22] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.
- [23] I. Rezek and S. J. Roberts. Estimation of coupled hidden Markov models with application to biosignal interaction modelling. In *Proc. IEEE Int. Conf. on Neural Network for Signal Processing*, volume 2, pages 804–813, 2000.
- [24] F. Sun and G. Hu. Speech recognition based on genetic algorithm for training HMM. *Electronics Letters*, 34(16):1563–1564, August 1998.
- [25] S. J. Young. Competitive training: A connectionist approach to the discriminative training of hidden Markov models. *IEE Proceedings - I*, 138(1):61–68, February 1991.