A Discriminative Model Corresponding to Hierarchical HMMs

Takaaki Sugiura, Naoto Gotou, and Akira Hayashi

Graduate School of Information Sciences, Hiroshima City University,
3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194, Japan
takaaki@robotics.im.hiroshima-cu.ac.jp

Abstract. Hidden Markov Models (HMMs) are very popular generative models for sequence data. Recent work has, however, shown that on many tasks, Conditional Random Fields (CRFs), a type of discriminative model, perform better than HMMs. We propose Hierarchical Hidden Conditional Random Fields (HHCRFs), a discriminative model corresponding to hierarchical HMMs (HHMMs). HHCRFs model the conditional probability of the states at the upper levels given observations. The states at the lower levels are hidden and marginalized in the model definition. We have developed two algorithms for the model: a parameter learning algorithm that needs only the states at the upper levels in the training data and the marginalized Viterbi algorithm, which computes the most likely state sequences at the upper levels by marginalizing the states at the lower levels. In an experiment that involves segmenting electroencephalographic (EEG) data for a Brain-Computer Interface, HHCRFs outperform HHMMs.

1 Introduction

Hidden Markov Models (HMMs) are very popular generative models for sequence data. Recent work has, however, shown that Conditional Random Fields (CRFs), a type of discriminative model, perform better than HMMs on many tasks [1].

There are several differences between CRFs and HMMs. (1) HMMs are generative models and thus model the joint probability of input (i.e., observations) and output data (i.e., states), whereas CRFs are discriminative models that model the conditional probability of output data given the input data. (2) HMMs make independence assumptions on observations given states, whereas CRFs do not. (3) For model parameter estimation, HMMs do not need the states, whereas CRFs do.

Hierarchical HMMs (HHMMs) are a generalization of HMMs with a hierarchical structure [2]. Murphy [3] has shown that HHMMs are a special kind of Dynamic Bayesian Networks (DBNs), and has derived an efficient inference algorithm [3].

In this paper, we propose the Hierarchical Hidden CRF (HHCRF), a discriminative model that corresponds to the HHMM, a generative model. In hierarchical models, we are mainly interested in the states at the upper levels, and hence, HHCRFs model the conditional probability of the states at the upper levels given observations. The states at the lower levels are hidden, and marginalized in the model definition.

We have developed two algorithms for the model. One is a parameter learning algorithm that needs only the states at the upper levels in the training data. Note that we need all the states to train standard CRF models. The other algorithm is the marginalized Viterbi algorithm, which computes the most likely state sequences at the upper levels by marginalizing the states at the lower levels. Note that a direct extension of the well known Viterbi algorithm computes the most likely joint sequence at the upper and lower levels, which is different from the sequence computed by the marginalized Viterbi algorithm.

In an experiment that involves segmenting electroencephalographic (EEG) data for a Brain-Computer Interface, HHCRFs outperform HHMMs.

2 Related Work

HHMMs were originally defined by Fine *et al.* [2]. Later, Murphy and Paskin [3] devised a DBN representation for HHMMs and a linear time inference algorithm. We have developed a semi-supervised learning algorithm for HHMMs [4]. Applications of HHMMs include handwritten character recognition, information extraction, video structure discovery, and topic transition detection.

CRFs were originally proposed by Lafferty *et al.* [1]. Since then, CRFs have successfully been applied to many problems including parsing, named entity recognition, object recognition, and activity recognition.

Sutton *et al.* [5] proposed Dynamic CRFs (DCRFs), an extension of CRFs, corresponding to factorial HMMs. Liao *et al.* [6] proposed hierarchical CRFs, another extension of CRFs, corresponding to HHMMs. Unfortunately, hierarchical CRFs do not have hidden states, hence all the states must be labeled for model parameter estimation.

Gunawardana *et al.* [7] proposed Hidden CRFs (HCRFs) for phone classification. HCRFs have two advantages over the previous CRF extensions. First, they have output feature functions which can express continuous Gaussian outputs. This is in contrast to binary valued output feature functions for most CRFs. The second advantage is that HCRFs have hidden states. HCRFs have 2 levels. The states at the bottom level are hidden and thus do not need to be labeled for training. The problem with HCRFs is that the states at the top level do not change with time. Therefore, whilst HCRFs can be applied to sequence classification, they cannot be applied to sequence segmentation or sequence labeling.

We were influenced by the above-mentioned approaches, and our HHCRFs share many characteristics with them. In HHCRFs, however, not only are the states at the lower levels hidden, but the states at the upper levels also change with time. This makes it possible to apply HHCRFs to sequence segmentation and sequence labeling without labeling the states at the lower levels for training.

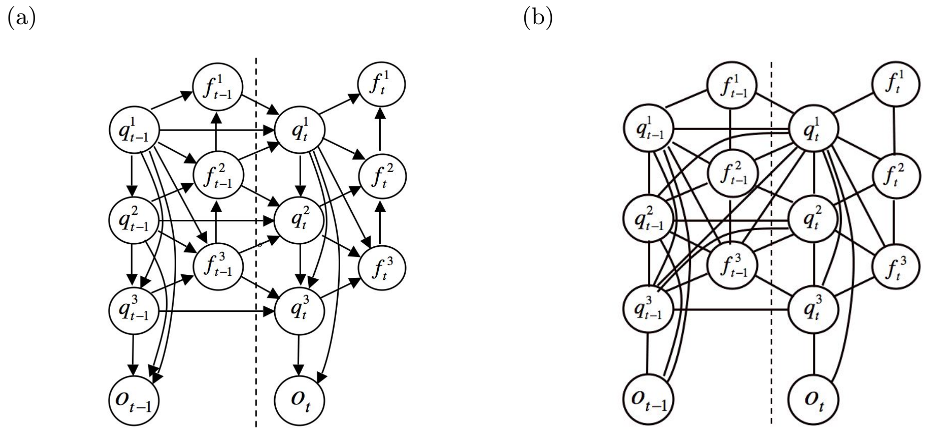

Fig. 1. (a) An HHMM represented as a DBN. (b) An HHCRF represented as an undirected graph. Both (a) and (b) describe only the part of the model from $t-1$ to t.

3 HHMMs

Hierarchical HMMs (HHMMs) are a generalization of HMMs with a hierarchical structure [2]. HHMMs have three kinds of states: internal, production, and end states. They also have three kinds of transitions: vertical, horizontal, and forced transitions. Murphy [3] has shown that an HHMM is a special kind of DBN, and has derived an efficient inference algorithm [3]. In what follows, we show how to represent an HHMM as a DBN.

3.1 Representing an HHMM as a DBN

We can represent an HHMM as a DBN as shown in Fig. 1(a). (We assume for simplicity that all production states are at the bottom of the hierarchy.) A state of the HHMM is denoted by q_t^d ($d \in \{1, \ldots, D\}$), where d is the hierarchy index: the top level has $d = 1$, and the bottom level has $d = D$.

f_t^d is the indicator variable which is equal to 1, if q_t^d has transited to its end state, otherwise it is 0. Note that if $f_t^d = 1$, then $f_t^{d'} = 1$ for all $d' > d$; hence the number of indicator variables that are equal to 0 denotes the level of the hierarchy we are currently on. The indicator variables play an important role in representing the HHMM as a DBN.

Defined below are the transition and output probability distributions. These complete the definition of the model. When q_t^d has transited to its end state, $f_t^d = 1$. This is the signal that the states at the upper levels can be changed. Furthermore, it is a signal that the next value of q_{t+1}^d should be determined by a vertical transition, instead of a horizontal transition. Formally, we denote these as follows:

$$
\begin{aligned}
p(q_t^d = j' | q_{t-1}^d = j, f_{t-1}^{d+1} = b, f_{t-1}^d = f, q_t^{1:d-1} = i) &= \begin{cases} \delta(j, j') & \text{if } b = 0 \\ A_i^d(j, j') & \text{if } b = 1 \text{ and } f = 0 \\ \pi_i^d(j') & \text{if } b = 1 \text{ and } f = 1 \end{cases} \\
p(f_t^d = 1 | q_t^d = j, q_t^{1:d-1} = i, f_t^{d+1} = b) &= \begin{cases} 0 & \text{if } b = 0 \\ Ae^d(i, j) & \text{if } b = 1 \end{cases} \\
\mathrm{E}[o_t | q_t^{1:D} = i] &= \mu(i) \\
\mathrm{Cov}[o_t | q_t^{1:D} = i] &= \sigma^2(i)
\end{aligned} \tag{1}
$$

where the state vector $q_t^{1:d} = \{q_t^1, \ldots, q_t^d\}_{d \in \{1, \ldots, D\}}$ is represented by an integer i (i.e. i is the index for "mega state"). In Eq. (1), we assume the dummy state $q_t^0 = 0$ (i.e. the root state) for notational convenience. We also assume dummy indicator variables $f_0^{2:D} = 1$ and $f_t^{D+1} = 1$ for the first slice and for the bottom level, respectively.

$\delta(j, j')$ is Kronecker's delta. $A_i^d(j, j')$ is the horizontal transition probability into the j'th state (except into an end state) from the jth state at level d. $\pi_i^d(j')$ is the vertical transition probability into the j'th state from the ith state at level d. $Ae^d(i, j)$ is the horizontal transition probability into an end state from the jth state at level d.

μ and σ^2 are the mean vector and covariance , respectively, of the Gaussian distribution of the observations emitted at each time. Note that for simplicity, we use scalar observations and a single Gaussian density output. We could of course, use the vector valued observations and a Gaussian mixture density output.

4 HHCRFs

4.1 Model

HHCRFs are undirected graphical models (as shown in Fig. 1(b)) which encode the conditional probability distribution:

$$
p(Q^1 | O; \Lambda) = \frac{1}{Z(O; \Lambda)} \sum_{Q^{2:D}} \sum_{F^{1:D}} \exp\left(\sum_{k=1}^{K} \lambda_k \Phi_k(Q^{1:D}, F^{1:D}, O) \right) \tag{2}
$$

where $Q^1 = \{q_1^1, \ldots, q_T^1\}$ is the state sequence at the top level [1], $O = \{o_1, \ldots, o_T\}$ is the sequence data (observations) and $\Lambda = \{\lambda_1, \ldots, \lambda_K\}$ is the model parameter. We represent the state sequence $Q^{1:D} = \{Q^1, \ldots, Q^D\}$ and the indicator variable sequence $F^{1:D} = \{F^1, \ldots, F^D\}$. $Z(O; \Lambda)$ is the partition function that ensures that $p(Q^1 | O; \Lambda)$ is properly normalized.

$$
Z(O; \Lambda) = \sum_{Q^1} \sum_{Q^{2:D}} \sum_{F^{1:D}} \exp\left(\sum_{k=1}^{K} \lambda_k \Phi_k(Q^{1:D}, F^{1:D}, O) \right) \tag{3}
$$

[1] For simplicity, we assume that only the state sequence at the top level is not hidden. We could of course assume that the state sequences at multiple upper levels are not hidden.

$\Phi_k(Q^{1:D}, F^{1:D}, O)$ is a feature function that can be arbitrarily selected.

In order to compare the performance of HHCRFs with that of HHMMs, which have a Markov structure in the state sequence, we restrict the feature function as $\Phi_k(Q^{1:D}, F^{1:D}, O) = \sum_{t=1}^{T} \phi_k(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t)$ and make the model structure equivalent to that of the HHMMs. Each feature function $\phi_k(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t)$ is as follows.

$$\begin{aligned}
\phi_{j,j',i,d}^{(Hor)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \Big(\delta(q_{t-1}^{d} = j) \cdot \delta(q_t^{d} = j') \cdot \delta(q_t^{1:d-1} = i) \\
&\qquad \cdot\ \delta(f_{t-1}^{d+1} = 1) \cdot \delta(f_{t-1}^{d} = 0)\Big) \quad \forall_j, \forall_{j'}, \forall_i, \forall_d \\
\phi_{i,j',d}^{(Ver)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \Big(\delta(q_t^{d-1} = i) \cdot \delta(q_t^{d} = j') \\
&\qquad \cdot \delta(f_{t-1}^{d+1} = 1) \cdot \delta(f_{t-1}^{d} = 1)\Big) \quad \forall_i, \forall_{j'}, \forall_d
\end{aligned}$$

$$\begin{aligned}
\phi_{i,j,d}^{(End)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \Big(\delta(q_t^{1:d-1} = i) \cdot \delta(q_t^{d} = j) \\
&\qquad \cdot \delta(f_t^{d+1} = 1) \cdot \delta(f_t^{d} = 1)\Big) \quad \forall_i, \forall_j, \forall_d \\
\phi_i^{(Occ)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \delta(q_t^{1:D} = i) \quad \forall_i \\
\phi_i^{(M1)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \delta(q_t^{1:D} = i) \cdot o_t \quad \forall_i \\
\phi_i^{(M2)}(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t) &= \delta(q_t^{1:D} = i) \cdot o_t^2 \quad \forall_i
\end{aligned}$$

where $\delta(q = q')$ is equal to 1 when $q = q'$ and 0 otherwise. The first three feature functions are transition features. $\phi_{j,j',i,d}^{(Hor)}$ counts the horizontal transition into the j'th state (except into an end state) from the jth state at level d. $\phi_{i,j',d}^{(Ver)}$ counts the vertical transition into the j'th state from the ith state at level d. $\phi_{i,j,d}^{(End)}$ counts the horizontal transition into an end state from the jth state at level d. $\phi_i^{(Occ)}, \phi_i^{(M1)}, \phi_i^{(M2)}$ are output features which are necessary to represent the Gaussian density output [7].

It can be shown that setting the parameter Λ (i.e., the weight of the feature functions) as follows gives the conditional probability distribution induced by HHMMs with the transition probability distributions and the output probability distributions defined in Eq. (1):

$$\begin{aligned}
\lambda_{j,j',i,d}^{(Hor)} &= \log A_i^d(j, j') & \lambda_i^{(Occ)} &= -\frac{1}{2}\left(\log 2\pi\sigma^2(i) + \frac{\mu^2(i)}{\sigma^2(i)}\right) \\
\lambda_{i,j',d}^{(Ver)} &= \log \pi_i^d(j') & \lambda_i^{(M1)} &= \frac{\mu(i)}{\sigma^2(i)} \qquad (4) \\
\lambda_{i,j,d}^{(End)} &= \log Ae^d(i, j) & \lambda_i^{(M2)} &= -\frac{1}{2\sigma^2(i)}
\end{aligned}$$

4.2 Parameter Estimation

Just as in HHMMs, parameter estimation for HHCRFs is based on the maximum likelihood principle given a training set $\mathcal{D} = \{O^{(n)}, Q^{1^{(n)}}\}_{n=1}^{N}$. The difference is that we maximize the conditional probability distribution $p(Q^1|O; \Lambda)$ for HHCRFs, whereas we maximize the joint probability distribution $p(Q^1, O; \Lambda_1)$ for HHMMs. Here, Λ_1 is the parameter for HHMMs. The conditional log-likelihood for HHCRFs is as follows.

$$\begin{aligned}\mathcal{L}(\Lambda) &= \sum_{n=1}^{N} \log p(Q^{1^{(n)}}|O^{(n)}; \Lambda) \\ &= \sum_{n=1}^{N} \log \left(\sum_{Q^{2:D}} \sum_{F^{1:D}} \exp \left(\sum_{k=1}^{K} \lambda_k \Phi_k(Q^{1^{(n)}}, Q^{2:D}, F^{1:D}, O^{(n)}) \right) \right) \\ &\quad - \sum_{n=1}^{N} \log Z(O^{(n)}; \Lambda) \end{aligned} \tag{5}$$

The gradient of Eq. (5), which is needed for estimating the parameter $\hat{\Lambda}$, is given by

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \lambda_k} =& \sum_{n=1}^{N} \sum_{Q^{2:D}} \sum_{F^{1:D}} \Phi_k(Q^{1^{(n)}}, Q^{2:D}, F^{1:D}, O^{(n)}) p(Q^{2:D}, F^{1:D}|Q^{1^{(n)}}, O^{(n)}; \Lambda) \\ & - \sum_{n=1}^{N} \sum_{Q^1} \sum_{Q^{2:D}} \sum_{F^{1:D}} \Phi_k(Q^{1:D}, F^{1:D}, O^{(n)}) p(Q^{1:D}, F^{1:D}|O^{(n)}; \Lambda) \end{aligned} \tag{6}$$

The right hand side of Eq. (6) is the difference between the expectation of feature values under the probability distribution $p(Q^{2:D}, F^{1:D}|Q^{1^{(n)}}, O^{(n)}; \Lambda)$ and that under $p(Q^{1:D}, F^{1:D}|O^{(n)}; \Lambda)$. Since $\Phi_k(Q^{1:D}, F^{1:D}, O) = \sum_{t=1}^{T} \phi_k(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}, o_t)$, the sufficient statistics to compute the first expectation are the transition probabilities $\{p(q_{t-1}^{2:D}, q_t^{2:D}, f_{t-1}^{1:D}, f_t^{1:D}|Q^{1^{(n)}}, O^{(n)}; \Lambda)|1 \leq t \leq T\}$ and the occupancy probabilities $\{p(q_t^{2:D}, f_t^{1:D}|Q^{1^{(n)}}, O^{(n)}; \Lambda)|1 \leq t \leq T\}$. Note that the state sequences are partially labeled because Q^1 is given. These probabilities can be computed using the junction tree algorithm [8], or by converting the hierarchical model to a flat model with mega states and applying the backward-forward-backward algorithm [9]. (We use the latter method in our experiment.) Here, the backward-forward-backward algorithm is an extension of the standard forward-backward algorithm to partially labeled state sequences.

The sufficient statistics to compute the second expectation are the transition probabilities $\{p(q_{t-1}^{1:D}, q_t^{1:D}, f_{t-1}^{1:D}, f_t^{1:D}|O^{(n)}; \Lambda)|1 \leq t \leq T\}$ and the occupancy probabilities $\{p(q_t^{1:D}, f_t^{1:D}|O^{(n)}; \Lambda)|1 \leq t \leq T\}$, which can be computed using the junction tree algorithm, or by converting the hierarchical model to a flat model with mega states and applying the forward-backward algorithm. (Once again, we use the latter method in our experiment.)

4.3 Marginalized Viterbi Algorithm for HHMMs

The well-known Viterbi algorithm can be used to compute the most likely mega state sequence $[\hat{Q}^{1:D}, \hat{F}^{1:D}] = \operatorname{argmax}_{Q^{1:D}, F^{1:D}} p(Q^{1:D}, F^{1:D} | O; \hat{\Lambda}_1)$. On the other hand, our marginalized Viterbi algorithm computes the most likely upper level state sequence $[\hat{Q}^1, \hat{F}^2]$ by marginalizing the states at the lower levels.

$$[\hat{Q}^1, \hat{F}^2] = \operatorname*{argmax}_{Q^1, F^2} p(Q^1, F^2 | O; \hat{\Lambda}_1) = \operatorname*{argmax}_{Q^1, F^2} \sum_{Q^{2:D}} \sum_{F^{3:D}} p(Q^{1:D}, F^{2:D}, O; \hat{\Lambda}_1) \quad (7)$$

We first explain the marginalized Viterbi algorithm for HHMMs and then for HHCRFs. The algorithm uses dynamic programming to obtain the most likely upper level state sequence $[\hat{Q}^1, \hat{F}^2]$.

Initialize: t=1

$$\delta_1(i) = \log p(q_1^1 = i \ , \ f_1^2 = 1 \ , \ o_1 \ ; \ \hat{\Lambda}_1) \qquad \forall_i$$

Iterate: t=2,. . . ,T

$$\delta_t(i) = \max_{i', 1 \le \tau < t} \Big(\delta_\tau(i') + \log A_0^1(i', i) + \alpha_{\tau,t}(i) \Big) \qquad \forall_i$$

$$\psi_t(i) = \operatorname*{argmax}_{i', 1 \le \tau < t} \Big(\delta_\tau(i') + \log A_0^1(i', i) + \alpha_{\tau,t}(i) \Big) \qquad \forall_i$$

where

$$\alpha_{\tau,t}(i) = \log p(f^2_{\tau+1:t-1} = 0 \ , \ f_t^2 = 1 \ , \ o_{\tau+1:t} \mid q^1_{\tau+1} = i \ , \ f^2_\tau = 1 \ ; \ \hat{\Lambda}_1)$$

Traceback:

$$\hat{q}_T^1 = \operatorname*{argmax}_i \delta_T(i), \qquad t = T$$

$$\text{Iterate while } t > 0 : \quad 1) \ (\hat{q}^1_{t'}, t') = \psi_t(\hat{q}_t^1), \qquad 2) \ t \leftarrow t'$$

$\delta_t(i)$ is the maximum of the log-probability along a single top level path at time t, that generates the first t observations $o_{1:t}$ and ends in the top level state i. $\psi_t(i)$ contains the traceback information of the highest probability path, and records the previous top level state i' and its ending time τ. Thus, τ is the segmentation boundary time between the sub-sequence $o_{1:\tau}$ ending with the top level state i' and the sub-sequence $o_{\tau+1:t}$ generated by the top level state i. $\alpha_{\tau,t}(i)$ is the probability of sub-sequence $o_{\tau+1:t}$ generated by the top level state i.

The time complexity of the Viterbi algorithm is $O(T)$, whereas that of the marginalized Viterbi algorithm is $O(T^2)$. This is the cost for finding the most likely upper level state sequence.

4.4 Marginalized Viterbi Algorithm for HHCRFs

The marginalized Viterbi algorithm for HHCRFs is similar to that for HHMMs, with $\delta_1(i)$ replaced by $\log \sum_{q_1^{2:D}} \sum_{f_1^{3:D}} \exp\Big(\sum_{k=1}^{K} \lambda_k \phi_k(q_1^{1:D}, f_1^{2:D}, o_1) \cdot \delta(q_1^1 = i) \cdot \delta(f_1^2 = 1)\Big)$, the horizontal transition probability $\log A_0^1(i', i)$ replaced by $\lambda_{i',i,0,1}^{(Hor)}$, and $\alpha_{\tau,t}(i)$ replaced by $\log \sum_{q_{\tau+1:t}^{2:D}} \sum_{f_{\tau+1:t}^{3:D}} \exp\Big(\sum_{t'=\tau+1}^{t} \sum_{k=1}^{K} \lambda_k \phi_k(q_{t'-1}^{1:D}, q_{t'}^{1:D}, f_{t'-1}^{2:D}, f_{t'}^{2:D}, o_{t'}) \cdot \delta(f_\tau^2 = 1) \cdot \delta(f_{\tau+1:t-1}^2 = 0) \cdot \delta(f_t^2 = 1) \cdot \delta(q_{\tau+1}^1 = i)\Big) - \lambda_{0,i,1}^{(Ver)}$. Its time complexity is the same as that for HHMMs: $O(T^2)$.

5 Experiment

We compare the performance of HHCRFs with that of HHMMs in segmenting and labeling EEG time series data for a Brain-Computer Interface (BCI). As reported in literature, the HMM has outperformed Fisher's linear discriminant in a *synchronous* BCI experiment where segmented EEG data is classified [10]. However, HMMs do not perform better than static classifiers in *asynchronous* experiments, where non-segmented continuous EEG data is first windowed and then classified [11, 12]. It is difficult to identify the beginning and end of each mental task in asynchronous experiments. We get around this problem by using hierarchical dynamical models and by segmenting and labeling the entire EEG data at the same time without windowing.

In our experiment, we use the BCI Competition III Dataset V, which is characterized as a "multi class problem, continuous EEG" [13]. The data set contains data from 3 subjects during 4 sessions, each 4 minutes long [2]. The subjects perform one of three mental tasks for about 15 seconds and then switch randomly to another task at the operator's request.

Both the HHMMs and the HHCRFs have 2 levels: the top level has 3 states, each corresponding to a mental task, while the bottom level has 5 states with a single Gaussian density output. The parameters of the HHCRFs are initialized using the corresponding parameters of the HHMMs. For hidden top level state inference, which is necessary to estimate the mental task at each time, we use three algorithms: Forward-Backward (FB), Joint Viterbi (JV) [3], and Marginalized Viterbi (MV).

Table 1 shows the accuracy rates for labeling the EEG data (i.e., estimating the mental task) at each time. The data is labeled according to the inferred top level state at each time. The FB algorithm, which computes the most probable

[2] The data from the first 3 sessions is used as training data, whilst the data from the last session is used as test data.

[3] the Viterbi algorithm in the joint space of the top and bottom level states

Table 1. Accuracy rates for labeling the EEG data. Mean (%) with standard deviation (%) in brackets. Average of 10 runs.

subject	HHMM			HHCRF		
	FB	JV	MV	FB	JV	MV
#1	79.05 (0.00)	78.52 (0.88)	79.54 (0.00)	94.58 (4.80)	80.75 (6.36)	92.77 (5.31)
#2	61.58 (0.00)	52.27 (0.37)	52.07 (0.00)	70.17 (0.08)	51.80 (4.00)	47.15 (0.00)
#3	34.40 (0.00)	34.40 (0.00)	34.40 (0.00)	32.11 (0.00)	32.11 (0.00)	32.11 (0.00)
average	58.34	55.06	55.34	65.62	54.89	57.34

Table 2. Segmenting the EEG data for Subject #1. Top level state changes within ± 0.5 sec. of the true time of change are considered as *true-positive's*. Average of 10 runs.

	HHMM			HHCRF		
	FB	JV	MV	FB	JV	MV
Precision (%)	20.00	16.50	18.18	53.39	23.07	57.32
Recall (%)	37.50	25.00	25.00	52.50	25.00	55.00
F measure (%)	26.09	19.85	21.05	52.92	23.81	56.08

state given all observations, has a high accuracy rate for both HHMMs and HHCRFs [4] .

In Table 2, we evaluate the performance in segmenting the EEG data (i.e., detecting the change of mental task) for Subject #1. The data is segmented when the top level state changes. We can see that HHCRFs outperform HHMMs in segmenting the EEG data. See also Fig. 2.

We also evaluate the performance in estimating the *segment* sequence (i.e., the mental task *order* sequence). During 10 runs, the FB and MV algorithms for HHCRFs produced the correct segment sequence [5] 8 and 6 times, respectively, whereas HHMMs failed to produce the correct sequence.

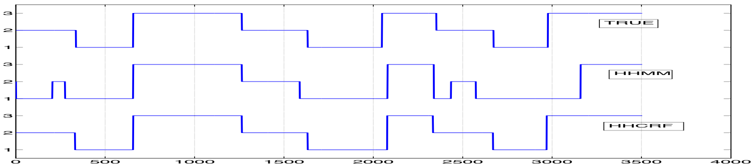

Fig. 2. Top level state sequence for Subject #1 as a function of time. **Top**: true sequence. **Middle**, and **Bottom**: The best HHMM and HHCRF sequences, respectively, in terms of the F measure.

[4] We found the estimation for Subject #3 extremely difficult, as was the case for all the submissions to the competition.

[5] i.e., $2 - 1 - 3 - 2 - 1 - 3 - 2 - 1 - 3$.

6 Conclusion

In this paper, we proposed HHCRFs, a discriminative model corresponding to the HHMM. We developed two algorithms for the model: a parameter learning algorithm that needs only the states at the upper levels in the training data, and the marginalized Viterbi algorithm which computes the most likely state sequences at the upper levels by marginalizing the states at the lower levels. In the experiment segmenting EEG data for a Brain-Computer Interface, the HHCRFs outperformed the HHMMs.

References

1. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th Int. Conf. Machine Learning. (2001)
2. Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden Markov model: Analysis and applications. Machine Learning **32**(1) (1998)
3. Murphy, K., Paskin, M.: Linear time inference in hierarchical HMMs. Advances in Neural Information Processing Systems **14** (2001)
4. Gotou, N., Hayashi, A., Suematsu, N.: Learning with segment boundaries for hierarchical HMMs. In: Proc. 3rd Int. Conf. Advances in Pattern Recognition. (2005)
5. Sutton, C., McCallum, A., Rohanimanesh, K.: Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. J. Mach. Learn. Res. **8**(Mar) (2007)
6. Liao, L., Fox, D., Kautz, H.: Hierarchical conditional random fields for GPS-based activity recognition. In: Proc. 12th Int. Symp. of Robotics Research. (2005)
7. Gunawardana, A., Mahajan, M., Acero, A., Platt, J.C.: Hidden conditional random fields for phone classification. In: Proc. Int. Conf. Speech Communication and Technology. (2005)
8. Huang, C., Darwiche, A.: Inference in belief networks: A procedural guide. Int. J. of Approximate Reasoning **15**(3) (1996)
9. Scheffer, T., Decomain, C., Wrobel, S.: Active hidden Markov models for information extraction. Lecture Notes in Computer Science **2189** (2001)
10. Obermaier, B., Guger, C., Neuper, C., Pfurtscheller, G.: Hidden markov models for online classification of single trial eeg data. Pattern Recogn. Lett. **22**(12) (2001) 1299–1309
11. Cincotti, F., et al.: Comparison of different feature classifiers for brain computer interfaces. In: Proc. 1st IEEE EMBS Conference on Neural Engineering. (2003) 645–647
12. Chiappa, S., Bengio, S.: Hmm and iohmm modeling of eeg rhythms for asynchronous bci systems. In: Proc. European Symposium on Artificial Neural Networks. (2004) 199–204
13. Blankertz, B., et al.: The bci competition iii: validating alternative approaches to actual bci problems. IEEE Transactions on Neural Systems and Rehabilitation Engineering **14**(2) (2006) 153–159