IDIAP RESEARCH REPORT

Sequence Classification with Input-Output Hidden Markov Models

Silvia Chiappa Samy Bengio

IDIAP–RR 04-13

August 2005

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 04-13

Sequence Classification with Input-Output Hidden Markov Models

Silvia Chiappa Samy Bengio

August 2005

e present a training and testing method for Input-Output Hidden Markov Model that is particularly suited for classification of sequences in which class information accumulates over time. We discuss two such cases: the discrimination of mental tasks from sequences of EEG features, common in Brain Computer Interface research, and phoneme classification from sequences of acoustic features for speech recognition. The objective function is modified so that training focuses on the improvement of classification accuracy. For both tasks the algorithm performs significantly better than the alternative solution proposed in the literature, specifically designed for other types of sequences.

1 Introduction

The Input-Output Hidden Markov Model (IOHMM), introduced by Bengio & Frasconi [1], is a graphical model which allows the mapping of input sequences into output sequences and thus provides a unified framework for solving three fundamental tasks of sequence processing, namely prediction, regression and classification. The model is trained to maximize the conditional distribution of an output sequence $\{y_1, \ldots, y_T\}$ given an input sequence $\{x_1, \ldots, x_T\}$, using the Expectation Maximization (EM) or Generalized EM (GEM) algorithm. Thus learning is supervised by the output sequence which represents the desired target.

For sequence classification, several scenarios have been reported in the literature. For instance, in [1] IOHMMs were applied to solve the 2-sequence problem, the parity problem and some grammatical inference tasks. In this case, class supervision was given at the end of the sequence only. During testing, the conditional probabilities $P(y_T|x_1, \ldots, x_T)$ were compared. In [4] the author applied IOHMMs to hand gesture recognition. In this case, a target sequence of identical class labels $\{y_1 = c, \ldots, y_T = c\}$ was assigned to the input sequence during training, while the same testing method of [1] was used.

In this work we focus the attention on the classification of sequences in which class information accumulates over time. The same target sequence configuration as in [4] is used. However, a testing approach which is consistent with training is selected. Furthermore the objective function is modified so that training focuses on the improvement of classification accuracy. As examples of tasks for which this algorithm is particularly suited, we present two sequence classification problems, namely the discrimination of mental tasks from sequences of EEG features and phoneme classification from sequences of acoustic features. We compare this algorithm with the alternative consistent solution found in the literature [1].

The rest of the paper is organized as follows. In Sec. 2 the IOHMM model is presented. Sec. 3 describes a training and testing algorithm commonly used for some sequence classification tasks. Sec. 4 introduces a new training and testing algorithm more adapted for other types of sequences. Sec. 5 describes the data and the experiments. Final conclusions are drawn in Sec. 6.

2 Input-Output Hidden Markov Models (IOHMMs)

An Input-Output Hidden Markov Model is an extended Hidden Markov Model (HMM) in which the distribution of the output variables $Y_{1:T} = \{Y_1, \ldots, Y_T\}$ and the hidden states $Q_{1:T} = \{Q_1, \ldots, Q_T\}$ are conditioned on a set of input variables $X_{1:T} = \{X_1, \ldots, X_T\}$ [1]. For classification, the input variables represent the observed sequences and the output variables represent the classes. As shown in Fig. 1, several conditional independence properties are assumed. In particular[1]:

$$P(q_t|q_{1:t-1}, x_{1:t}) = P(q_t|q_{t-1}, x_t)$$

and

$$P(y_t|q_{1:t}, y_{1:t-1}, x_{1:t}) = P(y_t|q_t, x_t)\,.$$

Thus to completely define an IOHMM we need the following set of distributions:

[1]We indicate with $P(Q_t = i)$ or $P(q_t)$ the probability that the variable Q_t takes the value i or $q_t \in [1, \ldots, N]$.We use the same notation for the output variables $Y_{1:T}$.

- the initial state probabilities $P(Q_1 = i|x_1)$,
- the state-transition probabilities $P(Q_t = i|Q_{t-1} = j, x_t)$,
- the emission probabilities $P(Y_t = c|Q_t = j, x_t)$.

For continuous inputs (the case assumed here), these distributions can be modeled by Multilayer Perceptron (MLP) [2] state networks N_j and output networks O_j, defined for each hidden state $j \in [1, \ldots, N]$. Each state network N_j predicts the next state distribution:

$$P(Q_t = i|Q_{t-1} = j, x_t)\,,$$

while each output network O_j computes the current class distribution:

$$P(Y_t = c|Q_t = j, x_t)\,.$$

We can observe that, while the HMM models the distribution of the observation sequences, the IOHMM models the conditional distribution of the desired output sequences given the observed input sequences. Thus, as opposed to the HMM framework for classification where for each class a different model is trained on examples of that class only, here a unique IOHMM model is trained, which yields more discriminant properties.

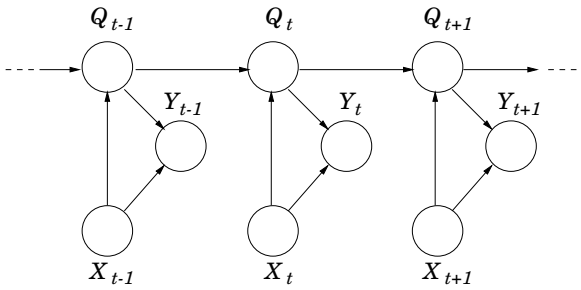


Figure 1: Graphical model specifying the conditional independence properties of an IOHMM. The nodes represent the random variables, while the arrows express direct dependencies between variables.

3 IOHMM Training and Testing Algorithms

One way of training an IOHMM model to classify sequences is to assign supervision, the class label, at the end of a sequence. In this case, training maximizes the data $\mathcal{D} = \{x^m_{1:T_m}, y^m_{T_m}\}_{m=1}^M$ log-likelihood:

$$\log L(\Theta; \mathcal{D}) = \log \prod_{m=1}^{M} P(y^m_{T_m}|x^m_{1:T}; \Theta) \tag{1}$$

of the M training sequences, using the Generalized EM algorithm (GEM) [5].
Like in the standard EM algorithm, at iteration p the E-step computes the conditional expectation of the complete data $\mathcal{D}_{comp} = \{x^m_{1:T_m}, y^m_{T_m}, q^m_{1:T_m}\}_{m=1}^M$ log-likelihood, given the data and the previous

parameters Θ^{p-1}, which can be expressed as (see [1] for details):

$$\begin{aligned} A(\Theta;\Theta^{p-1}) &= E[\log L_{comp}(\Theta;\mathcal{D}_{comp}))|\mathcal{D},\Theta^{p-1})] \\ &= \sum_{m=1}^{M}\sum_{i=1}^{N}\Big(\hat{g}^m_{i,T_m}\log P(y^m_{T_m}|Q_{T_m}=i,x^m_{T_m};\Theta) \\ &+ \sum_{t=1}^{T_m}\sum_{j=1}^{N}\hat{h}^m_{ij,t}\log P(Q_t=i|Q_{t-1}=j,x^m_t;\Theta)\Big)\,, \end{aligned}$$

where $\hat{g}_{i,T}$[2] and $\hat{h}_{ij,t}$ are the posterior probabilities of the state and transition distributions, computed with the old parameters:

$$\begin{aligned} \hat{g}_{i,T} &= P(Q_T=i|x_{1:T},y_T) \\ &= P(Q_T=i,y_T|x_{1:T})/P(y_T|x_{1:T}) \\ &= P(y_T|Q_T=i,x_T)P(Q_T=i|x_{1:T})/P(y_T|x_{1:T}) \end{aligned}$$

and

$$\begin{aligned} \hat{h}_{ij,t} &= P(Q_t=i,Q_{t-1}=j|x_{1:T},y_T) \\ &= P(Q_t=i,Q_{t-1}=j,y_T|x_{1:T})/P(y_T|x_{1:T}) \\ &= P(y_T|Q_t=i,x_{t+1:T})P(Q_t=i|Q_{t-1}=j,x_t)P(Q_{t-1}=j|x_{1:t-1})/P(y_T|x_{1:T})\,. \end{aligned}$$

In the above formulas:

$$P(y_T|x_{1:T}) = \sum_{i=1}^{N} P(y_T|Q_T=i,x_T)P(Q_T=i|x_{1:T})\,,$$

where $P(Q_t=i|x_{1:t})$ can be computed by backward recursion:

$$P(Q_t=i|x_{1:t}) = \sum_{j=1}^{N} P(Q_t=i|Q_{t-1}=j,x_t)P(Q_{t-1}=j|x_{1:t-1})\,;$$

while $P(y_T|Q_t=i,x_{t+1:T})$ can be computed by forward recursion:

$$P(y_T|Q_t=i,x_{t+1:T}) = \sum_{j=1}^{N} P(y_T|Q_{t+1}=j,x_{t+2:T})P(Q_{t+1}=j|Q_t=i,x_{t+1})\,.$$

In the M-step, the standard EM algorithm finds a value of Θ^p such that:

$$\Theta^p = \arg\max_{\Theta} A(\Theta;\Theta^{p-1})\,,$$

while, in our case, the MLP networks are trained to find a value Θ^p such that:

$$A(\Theta^p;\Theta^{p-1}) \geq A(\Theta^{p-1};\Theta^{p-1})\,. \tag{2}$$

using the standard gradient ascent formula:

$$\Theta^p = \Theta^{p-1} + \lambda\frac{\partial A(\Theta;\Theta^{p-1})}{\partial\Theta}\,,$$

[2]To simplify the notation, we omit the superscript and subscript m.

where λ is the learning rate parameter. Inequality (2) ensures that the log-likelihood is not decreased at each iteration. Indeed, it can be shown that:

$$\log L(\Theta^p; \mathcal{D}) - \log L(\Theta^{p-1}; \mathcal{D}) \geq A(\Theta^p; \Theta^{p-1}) - A(\Theta^{p-1}; \Theta^{p-1}) \,. \tag{3}$$

Once the IOHMM model has been trained, an unknown test sequence is assigned to the class with highest conditional probability, given the input sequence:

$$c^* = \arg\max_c P(Y_T = c | x_{1:T}) \,. \tag{4}$$

4 Alternative Training and Testing Algorithms

The algorithm described in Sec. 3 has been used for classifying sequences in which the value of a single new input may change the classification result based on the previous part of the sequence [1]. One such example is the parity problem, where the objective is to determine whether a binary sequence contains an odd or even number of ones. This problem can be solved, for example, by an IOHMM with two fully-connected hidden states which approximates the following rules:

$$\begin{cases} q_1 = 0 & \text{if } x_1 = 0 \\ q_1 = 1 & \text{if } x_1 = 1 \\ q_t = q_{t-1} & \text{if } x_t = 0,\ t > 1 \\ q_t \neq q_{t-1} & \text{if } x_t = 1,\ t > 1 \end{cases} \qquad \begin{cases} P(y_T = 0/1 | q_t = 0, x_t = 0) = 1/0 \\ P(y_T = 0/1 | q_t = 1, x_t = 0) = 0/1 \\ P(y_T = 0/1 | q_t = 0, x_t = 1) = 0/1 \\ P(y_T = 0/1 | q_t = 1, x_t = 1) = 1/0 \,, \end{cases}$$

that is, the value of the state q_t gives information about the parity of the previous part of the sequence $x_{1:t-1}$.

In other kinds of sequences, e.g. EEG sequences used in Brain Computer Interface research or speech sequences (see details of such data in Sec. 5), evidence about the class of the whole sequence accumulates over time. Furthermore, features from a window of raw data are usually extracted so that a single input already conveys some class information. In this case we may aid training by specifying the class label for each input of the sequence. The log-likelihood to maximize becomes:

$$\log L(\Theta; D) = \log \prod_{m=1}^{M} P(y^m_{1:T_m} | x^m_{1:T_m}; \Theta) \,. \tag{5}$$

The corresponding posterior probabilities $\hat{g}_{i,t}$ and $\hat{h}_{ij,t}$ can be computed in a similar way as described in Sec. 3. If we define:

$$\alpha_{i,t} = P(y_{1:t}, Q_t = i | x_{1:t})$$

and

$$\beta_{i,t} = P(y_{t+1:T} | Q_t = i, x_{t:T}) \,,$$

we can write:

$$\begin{aligned} \hat{g}_{i,t} &= P(Q_t = i | x_{1:T}, y_{1:T}) = P(Q_t = i, y_{1:T} | x_{1:T}) / P(y_{1:T} | x_{1:T}) \\ &= P(y_{1:t}, Q_t = i | x_{1:t}) P(y_{t+1:T} | Q_t = i, x_{t:T}) / P(y_{1:T} | x_{1:T}) \\ &= \alpha_{i,t} \beta_{i,t} / P(y_{1:T} | x_{1:T}) \,. \end{aligned}$$

and

$$\begin{aligned} \hat{h}_{ij,t} &= P(Q_t = i, Q_{t-1} = j | x_{1:T}, y_{1:T}) \\ &= P(Q_t = i, Q_{t-1} = j, y_{1:T} | x_{1:T}) / P(y_{1:T} | x_{1:T}) \\ &= P(y_t | Q_t = i, x_t) \alpha_{j,t-1} \beta_{i,t} P(Q_t = i | Q_{t-1} = j, x_{1:T}) / P(y_{1:T} | x_{1:T}) \end{aligned}$$

In the above formulas, $\beta_{i,t}$ can be computed by forward recursion:

$$\beta_{i,t} = \sum_{j=1}^{N} P(y_{t+1}|Q_{t+1} = j, x_{t+1}) P(Q_{t+1} = j|Q_t = i, x_{t+1}) \beta_{j,t+1} \,,$$

and

$$P(y_{1:T}|x_{1:T}) = \sum_{i=1}^{N} \alpha_{i,T} \,,$$

where $\alpha_{i,t}$ can be computed by backward recursion:

$$\alpha_{i,t} = P(y_t|Q_t = i, x_t) \sum_{j=1}^{N} P(Q_t = i|Q_{t-1} = j, x_t) \alpha_{j,t-1} \,.$$

Note that these equations are very similar to the classical HMM equations, a part from the conditioning on x_t.
This training strategy was followed by [4] for a problem of hand gesture recognition. However, for testing Eq. (4) was used, while, to be consistent, an unknown test sequence should be assigned to the class c^* such that:

$$c^* = \arg\max_c P(Y_1 = c, \ldots, Y_T = c|\Theta) \,. \tag{6}$$

Furthermore, when using (5) as objective function, training may focus on uninteresting dependencies in the data. Indeed, while in the algorithm described in Sec. 3 the number of possible outputs was given by the number of classes C, here we have C^T possible sequences of outputs: C consisting of identical elements and all the rest consisting of mixed elements. We can illustrate the consequences with the following example.
The task consists in discriminating among three mental tasks from the corresponding EEG sequences. Figure 2(a) shows the evolution of the log-likelihood with training iterations. The solid line indicates the log-likelihood to maximize during training, the dashed line indicates the log-likelihood of sequences consisting of identical but incorrect class labels, and the dotted line indicates the log-likelihood of sequences of mixed class labels. We used a configuration with three fully-connected hidden states. During training we present to the model only one type of input sequence whose elements have all the same class label. This strong characteristic is learned by the model. Indeed, after around 25 training iterations, the model starts associating each hidden state to a different class[3]. Furthermore, the transition matrices tend to reach a diagonal configuration. As a consequence, the likelihood of sequences of identical class labels, both correct and incorrect, tends to increase[4].

To avoid this problem, we can focus the training to discriminate only between joint probabilities of identical outputs, which are the ones associated to the training sequences and compared during testing, by defining the following objective function:

$$\log L_c(\Theta; \mathcal{D}) = \log \prod_{m=1}^{M} \frac{P(Y_1^m = c, \ldots, Y_{T_m}^m = c|x_{1:T_m}^m; \Theta)}{\sum_{i=1}^{C} P(Y_1^m = i, \ldots, Y_{T_m}^m = i|x_{1:T_m}^m; \Theta)} \,, \tag{7}$$

where c is the correct class label.
There are two problems in the maximization of $\log L_c(\Theta; \mathcal{D})$. First, we cannot derive a simple formula for the conditional expectation of the complete data log-likelihood. Second, inequality (3) does not hold anymore, that is the increasing of the expectation at each iteration would not imply the increasing

[3]That is, assuming, without loss of generality, that state 1 is associated to class 1, $P(Y_t = 1|q_t = 1, x_t) \to 1$ and $P(Y_t = 1|q_t \neq 1, x_t) \to 0$ for $x_t \in$ class 1.

[4]The same behavior would happen with a higher number of hidden states, while it would get reduced with less states, disappearing with one state, but then, no more temporal relation can be taken into account.

of $\log L_c(\Theta;\mathcal{D})$. The kind of auxiliary functions for which condition (3) is satisfied are called strong-sense auxiliary functions [7]. However, other kind of functions, called weak-sense auxiliary functions, have been successfully used for discriminative training of HMMs [7]. A functions $\mathcal{G}(\Theta;\Theta^{p-1})$ is a weak-sense auxiliary function for $\mathcal{F}(\Theta)$ if it satisfies the following property:

$$\left.\frac{\partial \mathcal{G}(\Theta;\Theta^{p-1})}{\partial \Theta}\right|_{\Theta=\Theta^{p-1}} = \left.\frac{\partial F(\Theta)}{\partial \Theta}\right|_{\Theta=\Theta^{p-1}} .$$

that is, if Θ^{p-1} is a stationary point of $\mathcal{G}(\Theta;\Theta^{p-1})$, it is also a stationary point of $\mathcal{F}(\Theta)$. Thus, if the GEM updates converge, it will be to a stationary of $\mathcal{F}(\Theta)$.
In our case, we define a new function $\mathcal{H}(\Theta;\Theta^{p-1};\mathcal{D}_{comp})$ as follows:

$$\begin{aligned}\mathcal{H}(\Theta;\Theta^{p-1};\mathcal{D}_{comp}) = & \\ & \sum_{m=1}^{M}\Big(\big(1-L_c(\Theta^{p-1};\mathcal{D})\big)\log L_{c,comp}(\Theta;\mathcal{D}_{comp}) \\ & -\sum_{i=1,i\neq c}^{C} L_i(\Theta^{p-1};\mathcal{D})\log L_{i,comp}(\Theta;\mathcal{D}_{comp})\Big) .\end{aligned}$$

The conditional expectation of $\mathcal{H}(\Theta;\Theta^{p-1};\mathcal{D}_{comp})$, given the data and the old parameters, is a weak-sense auxiliary function for (7), that is:

$$\left.\frac{\partial E[\mathcal{H}(\Theta;\Theta^{p-1};\mathcal{D}_{comp})|\mathcal{D};\Theta^{p-1})]}{\partial \Theta}\right|_{\Theta=\Theta^{p-1}} = \left.\frac{\partial \log L_c(\Theta;\mathcal{D})}{\partial \Theta}\right|_{\Theta=\Theta^{p-1}} .$$

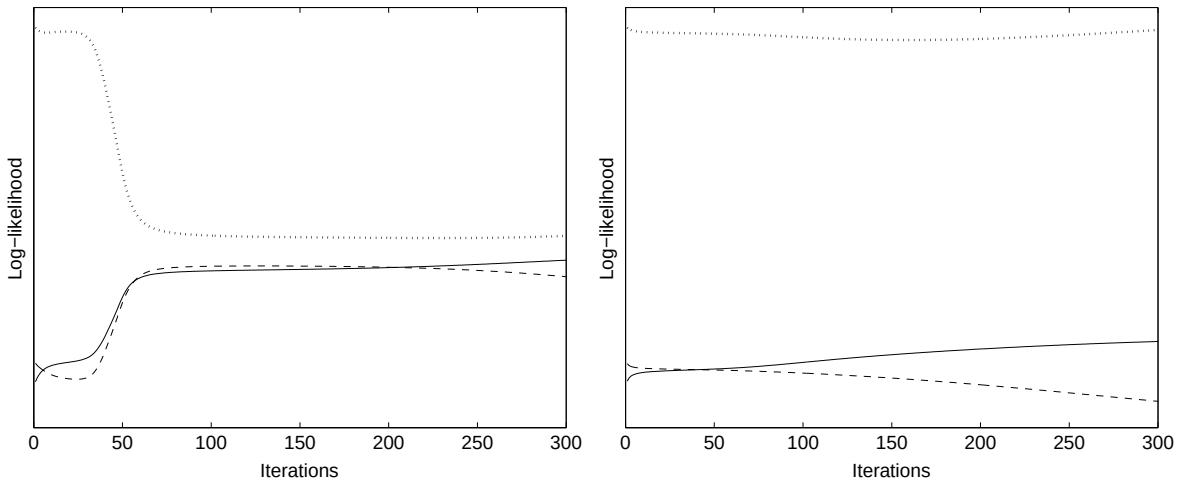
In all our experiments, the use of this auxiliary function solved the learning problem discussed above. Fig. 2(b) shows the behavior of the log-likelihood on the EEG example. The solid line indicates the log-likelihood (Eq. (5)) of sequences consisting of identical correct class labels, the dashed line indicates the log-likelihood of sequences consisting of identical but incorrect class labels, and the dotted line indicates the log-likelihood of sequences of mixed class labels. It can be seen that the ratio between the log-likelihood of sequences of identical, correct and incorrect, class labels, increases with training iterations, as desired.

5 Data and Experiments

We have evaluated the training and testing methods described in Sec. 3 (Eq. (1)-(4)) and Sec. 4 (Eq. (6)-(7)) on two sequence classification problems, namely the discrimination of mental tasks for asynchronous Brain Computer Interface (BCI) systems and the classification of phonemes for speech recognition.

5.1 EEG Data

The EEG potentials were recorded with a portable system using 32 electrodes located at standard positions of the 10-20 International System, at a sample rate of 512 Hz. The raw potentials (without artifact rejection or correction) were spatially filtered using a Common Average Reference filter [6]. Then the power spectral density was computed over half a second of data with a temporal shift of 250 milliseconds, in the band 8-30 Hz and for the following 19 electrodes: F3, FC1, FC5, T7, C3, CP1, CP5, P3, Pz, P4, CP6, Cp2, C4, T8, FC6, FC2, F4, Fz and Cz.
The data was acquired from two healthy subjects without any experience with BCI systems during three consecutive days. Each day, the subjects performed 5 recording sessions lasting 4 minutes. During each recording session the subjects had to concentrate on three different mental tasks: repetitive



(a) (b)

Figure 2: EEG data log-likelihoods evolution for different types of output sequences. (a): Unconstrained training algorithm. (b): Constrained training algorithm. Solid line(-): Output sequences of correct class labels. Dashed line (–): Output sequences of identical but incorrect class labels. Dotted line (.): Output sequences of mixed classes.

imagination of self-paced left and right hand movements and mental generation of words starting with a given letter. The subjects had to change every 20 seconds between one mental task and another under the instruction of an operator.

The IOHMM models were trained on the EEG signal of the first 2 days of recordings, while the last day was used as validation and test sets. The validation set was used to select the number of iterations for the GEM, the number of states (up to 6) and the number of hidden units (between 5 and 50) for the MLP transition and emission networks. The MLP networks had one hidden layer. The asynchronous protocol used to record the data makes impossible to determine the beginning and end of a single mental action. Given that and the lack of prior information about the dynamics of the EEG rhythms we used a fully connected topology in which each hidden state could be reached by any other state. We split each recording session into segments of signal lasting 2 seconds, with a shift of half a second, obtaining around 4000 training, 1000 validation and 1000 testing sequences.

Table 1 shows the classification error rate for the two subjects. In both cases, the constrained method described in Sec. 4 gives an improvement over the algorithm of Sec. 3. A standard test of difference of proportions reveals that this improvement is statistically significant with 99% of confidence.

	IOHMM Eq. (1)-(4)	IOHMM Eq. (6)-(7)
Subject A	34.8%	19.0%
Subject B	36.8%	18.5%

Table 1: Error rate of Subject A and Subject B. The first column gives the performance of the algorithm described in Sec. 3 (Eq. (1)-(4)), while the second column gives the performance of the algorithm of Sec. 4 (Eq. (6)-(7)).

5.2 Speech Data

The phoneme data was obtained from a pre-segmentation of the OGI Numbers 95 (N95) database [3]. The raw data was converted into a set of features based on Mel-Frequency Cepstrum Coefficients (MFCC) [8] (with 39 components, consisting of three groups of 13 coefficients, namely the static coefficients and their first and second derivatives).

Out of a total of 27, the five hardest (in terms of confusion matrices) phonemes were selected from a preliminary experiment. The selection resulted in 9274 training, 4636 validation and 5304 test phonemes sequences. The validation set was used to choose the number of iterations of the GEM and the number of hidden units (between 5 and 100) for the MLP transition and emission networks. The MLP networks had one hidden layer. We used the left-right topology with 3 hidden states, as commonly done in speech recognition.

The results are presented in Table 2, where we can observe that the constrained training algorithm of Sec. 4 gives statistically significant (99% of confidence) superior performance with respect to the algorithm described in Sec. 3.

	IOHMM Eq. (1)-(4)	IOHMM Eq. (6)-(7)
N95	13.9%	10.3%

Table 2: Error rate in the discrimination of 5 phonemes of the N95 database. The first column gives the performance of the algorithm described in Sec. 3 (Eq. (1)-(4)), while the second column gives the performance of the algorithm of Sec. 4 (Eq. (6)-(7)).

6 Conclusion

We have presented a training and testing method for classification of sequences using IOHMMs. The particular configuration proposed is based on the assumption that class information about the whole sequence accumulates over time. This is the case of many sequences used in real-word problems, like the examples discussed in the paper, namely the discrimination of mental tasks from EEG data for BCI research and phoneme classification for speech recognition. Here, in particular, features from raw data are extracted so that each element of the sequence conveys some information about the class. The objective function is modified so that training focuses on the improvement of classification performance. In both cases the algorithm performs better than another consistent solution proposed in the literature, specifically designed for other types of sequences, showing that this method can be used as a valid alternative.

7 Acknowledgments

The authors would like to thank J. del R. Millán for fruitful discussions.
This work is supported by the Swiss National Science Foundation through the National Centre of Competence in Research on "Interactive Multimodal Information Management".

References

[1] Y. Bengio and P. Frasconi. Input-output HMMs for sequence processing. *IEEE Transactions on Neural Networks*, 7:1231–1249, 1996.

[2] C Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] R.A. Cole, K. Roginski, and M. Fanty. The OGI numbers database. Technical report, Oregon Graduate Institute, 1995.

[4] S. Marcel, O. Bernier, and J.-E. Viallet et D. Collober. Hand gesture recognition using input-ouput hidden Markov models. In *Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition*, pages 456–461, 2000.

[5] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley and Sons, 1997.

[6] P. L. Nunez. *Neocortical Dynamics and Human EEG Rhythms*. Oxford University Press, 1995.

[7] D. Povey, P. C. Woodland, and M. J. F. Gales. Discriminative MAP for acoustic model adaptation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 312–315, 2003.

[8] L.R. Rabiner and B-H Juang. *Fundamentals of Speech Recognition*. PTR Prentice-Hall, Inc., 1993.