Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition Letters 24 (2003) 1395–1407

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A sequential pruning strategy for the selection of the number of states in hidden Markov models

Manuele Bicego [a,*], Vittorio Murino [a], Mário A.T. Figueiredo [b]

[a] *Department of Computer Science, University of Verona, Ca' Vignal 2, Strada Le Grazie 15, 37134 Verona, Italy*
[b] *Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal*

Received 22 February 2002; received in revised form 16 September 2002

Abstract

This paper addresses the problem of the optimal selection of the structure of a hidden Markov model. A new approach is proposed, which is able to deal with drawbacks of standard general purpose methods, like those based on the Bayesian inference criterion, i.e., computational requirements, and sensitivity to initialization of the training procedures. The basic idea is to perform "decreasing" learning, starting each training session from a "nearly good" situation, derived from the result of the previous training session by pruning the "least probable" state of the model. Experiments with real and synthetic data show that the proposed approach is more accurate in finding the optimal model, is more effective in classification accuracy, while reducing the computational burden.

Keywords: Hidden markov models; Model selection; Bayesian inference criterion; Minimum description length; State pruning

1. Introduction

The hidden Markov model (HMMs) approach (Rabiner, 1989) is a widely used method for probabilistic sequence modelling. Although the basic theory and tools were developed by Baum et al. in the late 1960s (Baum et al., 1970; Baum, 1970), HMMs have only been extensively applied in the last decade. Speech recognition (Rabiner, 1989), handwritten character recognition (Hu et al., 1996), DNA and protein modelling (Hughey and Krogh, 1996), gesture recognition (Eickeler et al., 1998) and behavior analysis and synthesis (Jebara and Pentland, 1999), are examples of problems in which HMMs have been exploited.

A practical but fundamental issue to be addressed when using HMMs is the determination of its structure, namely the topology and the number of states. The former aspect regards the possibility of introducing some constraints in the HMM structure, such as forcing the presence or absence of connections between certain states; the latter issue is directly addressed in this paper, and concerns the determination of the number of states. Although some special purpose approaches have been proposed (e.g. Stolcke and Omohundro, 1993;

* Corresponding author.
E-mail addresses: bicego@sci.univr.it (M. Bicego), vittorio.murino@univr.it (V. Murino), mtf@lx.it.pt (M.A.T. Figueiredo).

0167-8655/03/$ - see front matter
PII: S0167-8655(02)00380-X

Brand, 1999; Bicego et al., 2001), the typical solution is to use some heuristics, or some general purpose model selection method which is not specifically oriented to HMMs. Cross validation (CV) (Stone, 1974) is one such method, also employed in other context to obtain statistically reliable evaluation of system performances; this method is computationally heavy and does not use the available data efficiently. In CV, the observed data is split in two subsets: one becomes the *training set*, the other is called *test set* (the splitting strategy depends on the specific details of the CV technique chosen); different models are then obtained using only the training set (e.g., varying the model structure) and the one showing best performance on the test set is chosen. Other model selection methods which can be used for HMMs include the minimum description length (MDL) principle (Rissanen, 1986), the Bayesian inference criterion (BIC) (Schwarz, 1978), and the MML criterion (Oliver et al., 1996). These methods address the model selection problem by training several models, with different structures, and then choosing the one that maximizes a certain selection criterion. These approaches perform rather accurately, allowing an increase in performance (e.g., Li et al., 2001; Raftery, 1995; and Zimmermann and Bunke, 2001). Although these techniques are less computationally expensive than CV, they still involve a considerable computational burden, since one full training is required for each candidate model structure.

Another problem, common to all these approaches, is the local/greedy behavior of the standard algorithm used to estimate the HMM parameters from training data, i.e., the expectation-maximization (EM) algorithm (Dempster et al., 1977). This learning procedure, starting from some initial estimate, converges to the nearest local maximum of the likelihood function. Therefore, the initialization crucially affects the obtained model estimate, since the likelihood function is highly multi-modal, and this behavior strongly affects the model order selection criteria. A typical solution, used for discrete HMM but deleterious for continuous HMMs, is to use several random initializations and choose as final estimate the one with the highest likelihood. Other clever approaches, like preliminary clustering of coefficients, can also be used (e.g., Bicego and Murino, submitted for publication).

In this paper a new approach is proposed, which simultaneously addresses the two issues mentioned above: the computational burden of model selection, and the initialization phase. The key idea is to use a decreasing learning strategy, starting each training session from an informative situation derived from the previous training phase. More specifically, we propose a procedure which consists in starting the model training using a large number of states, run the estimation algorithm, and, after convergence, evaluate the chosen model selection criterion for that model. Then, the "least probable" state is pruned, and this configuration is taken as initial situation from which to start again the training procedure. In this way, each training session is started from a "nearly good" estimate. A related approach has been successfully used for Gaussian mixtures in (Figueiredo et al., 1999). The key observation supporting this approach is that, when the number of states is extremely large, the initialization dependency of the estimate is much weaker than when the number of states is close to the optimum. Moreover, the "good" initialization drastically reduces the number of iterations required by the learning algorithm, resulting in a less computational demanding procedure. The idea of pruning model selection was successfully employed also in the field of Neural Networks (see Bishop, 1995 and the references herein contained). The proposed method could be applied for all types of HMMs, discrete, continuous, autoregressive and so on. Moreover, it can be used with any model selection criterion: in this paper we consider BIC (Schwarz, 1978), and the mixture minimum description length (MMDL), a criterion proposed in (Figueiredo et al., 1999) for Gaussian mixtures and here extended to HMMs. It is worth noting that although Gaussian mixtures can be considered as (simple) special cases of HMMs, applying MMDL and the pruning strategy to HMMs involves additional conceptual and technical difficulties which will be addressed in this paper.

In the experimental session, the pruning and the normal strategy are compared in terms of accuracy of model selection, classification performance, and computational requirements, using both real and

synthetic data. We show that our approach is more accurate in finding the optimal model, more effective in classification accuracy, while exhibiting a lower computational burden.

The rest of the paper is organized as follows. Section 2 presents a brief introduction to HMMs, mainly to setup the notation used throughout the paper. In Section 3, several model selection criteria are described, including our adaptation of the MMDL method to HMMs. The proposed technique is described in Section 4. Experiments and results are reported in Section 5, and, finally, conclusions and future perspectives are addressed in Section 6. Appendix A contains a brief review of the concept of stationary distribution, while Appendix B contains the proof of the equivalence between HMMs with more than one Gaussian per state and HMMs with only one Gaussian per state, which plays a central role in our approach.

2. Hidden Markov models

A discrete-time first-order HMM (Rabiner, 1989) is a probabilistic model that describes a stochastic sequence $\boldsymbol{O} = O_1, O_2, \ldots, O_T$ as being an indirect observation of an underlying (hidden) random sequence $\boldsymbol{Q} = Q_1, Q_2, \ldots, Q_T$, where this hidden process is Markovian, though the observed process may not be so. A discrete HMM is formally defined by the following elements:

- A set $S = \{S_1, S_2, \ldots, S_k\}$ of (hidden) states.
- A transition matrix $\boldsymbol{A} = \{A_{ij} = A(S_i \to S_j)\}$, of size $k \times k$, where element $A(S_i \to S_j) \geqslant 0$ is the probability of going from state S_i to state S_j:

$$\begin{aligned} A_{ij} &= A(S_i \to S_j) \\ &= P[Q_{t+1} = S_j | Q_t = S_i], \quad 1 \leqslant i, j \leqslant k \end{aligned} \tag{1}$$

  where Q_t denotes the state occupied by the system at time t. Since $\sum_{j=1}^{k} A_{ij} = 1$, $\boldsymbol{A}$ is called a *stochastic matrix*. We will consider only *stationary* HMMs, i.e., the transition matrix does not depend on t.
- A set $V = \{v_1, v_2, \ldots, v_m\}$ of observation symbols.
- An emission matrix $\boldsymbol{B} = \{b(v_j|S_i)\}$ (of size $k \times m$) indicating the probability of observing symbol v_j from state S_i, that is,

$$b(v_j|S_i) = P[O_t = v_j | Q_t = S_i], \quad 1 \leqslant i \leqslant k, \ 1 \leqslant j \leqslant m \tag{2}$$

  with $b(v_j|S_i) \geqslant 0$ and, naturally, $\sum_{j=1}^{m} b(v_j|S_i) = 1$.
- An initial state probability distribution $\boldsymbol{\pi} = \{\pi(S_i)\}$,

$$\pi(S_i) = P[q_1 = S_i], \quad 1 \leqslant i \leqslant k, \tag{3}$$

  with, $\pi(S_i) \geqslant 0$, and $\sum_{i=1}^{k} \pi(S_i) = 1$.

An HMM is completely specified by a five-tuple $\boldsymbol{\lambda} = (S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ and defines a joint probability distribution on the space of hidden and observed sequences, i.e., $P(\boldsymbol{O} = \boldsymbol{o}, \boldsymbol{Q} = \boldsymbol{q}|\boldsymbol{\lambda})$.

There are three main problems involved with using HMMs:

(1) Given the HMM $\boldsymbol{\lambda} = (S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, we want to compute the marginal probability $P(\boldsymbol{O} = \boldsymbol{o}|\boldsymbol{\lambda})$, usually called the *likelihood function*, i.e., the probability that an observed sequence $\boldsymbol{o} = o_1, o_2, \ldots, o_T$ (with $o_t \in V$, for $t = 1, 2, \ldots, T$) was generated by the model $\boldsymbol{\lambda}$. This is usually solved by the so-called *forward-backward procedure* (Baum, 1970).

(2) Given $\boldsymbol{\lambda} = (S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, and an observed sequence $\boldsymbol{o} = o_1, o_2, \ldots, o_T$, we want to determine the state sequence that most probably generated $\boldsymbol{o}$, that is, $\hat{\boldsymbol{q}} = \hat{q}_1, \hat{q}_2, \ldots, \hat{q}_T$ (with $1 \leqslant \hat{q}_t \leqslant N$), such that

$$\hat{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} \boldsymbol{P}(\boldsymbol{O} = \boldsymbol{o}, \boldsymbol{Q} = \boldsymbol{q}|\boldsymbol{\lambda}).$$

  This problem is solved by the *Viterbi algorithm* (Forney, 1973).

(3) Given a set of L observed sequences $\mathcal{O} = \{\boldsymbol{o}^{(l)}\}$, where $1 \leqslant l \leqslant L$, and $\boldsymbol{o}^{(l)} = o_1, o_2, \ldots, o_{T_l}$, assumed to be independent samples from a common HMM $\boldsymbol{\lambda} = (S, V, \boldsymbol{A}, \boldsymbol{B}, \pi)$, we want to estimate $\boldsymbol{\lambda}$. This is usually obtained by adopting the maximum likelihood (ML) criterion, that is,

$$\begin{aligned} \hat{\boldsymbol{\lambda}} &= \arg\max_{\boldsymbol{\lambda}} P(\mathcal{O}|\boldsymbol{\lambda}) \\ &= \arg\max_{\boldsymbol{\lambda}} \prod_{l=1}^{L} P(\boldsymbol{O} = \boldsymbol{o}^{(l)}|\boldsymbol{\lambda}); \end{aligned}$$

this problem is usually referred to as HMM *training*. The best-known way to implement this ML criterion is the Baum–Welch (BW) algorithm (Baum et al., 1970). BW is an instance of the well-known EM algorithm (Dempster et al., 1977) for ML estimation with missing data (here, the missing data is the hidden sequence $\boldsymbol{q}$).

In many applications, V is a continuous set (e.g., $V = \mathbb{R}$, or $V = \mathbb{R}^d$). In this case, instead of a matrix of symbol probabilities $\boldsymbol{B}$, for each state S_i we have an emission probability density function $b(o|S_i)$, for $o \in V$, and of course with $\int_V b(o|S_i)\,\mathrm{d}o = 1$. For real (scalar or vector) observations, a very common approach is to model $b(o|S_i)$ as a mixture of Gaussians,

$$b(o|S_i) = \sum_{j=1}^{M_i} c_{ij}\mathcal{N}\left(o|\theta_{ij}\right), \tag{4}$$

where $\mathcal{N}(o|\theta)$ denotes a Gaussian density with parameters (e.g., mean and covariance) θ. The observations from state S_i are modelled as samples from a Gaussian mixture with M_i components, with c_{ij} denoting the mixture coefficient (or weight) of the jth component in state S_i. In this mixture-based case, whose adaptation of the Baum–Welch procedure is straightforward (Juang et al., 1986), we let $\boldsymbol{B}$ denote the set of all the mixtures parameters (the M_i's, and the θ_{ij}'s) and an HMM is thus completely defined by $\boldsymbol{\lambda} = (S, \boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{B})$.

3. Model selection

A fundamental problem when using HMMs in practical applications is the determination of the number of states k, and/or, when using HMMs with emission densities modelled by Gaussian mixtures, the number of components at each state, $M_1, \ldots, M_k$. This is usually called the *model selection* problem, even if, for the sake of correctness, model selection involves the choice of both topology and number of states. However, the topology often depends on the specific application addressed, and typically fixed a priori; our method is general, not tailored to a particular application domain. It is well known that this problem cannot be addressed by the ML criterion (Rissanen, 1986). The reason lies in the fact that the models are nested, i.e., an HMM with fewer states, or fewer components in one or more states, can always be seen as a particular case of a larger model. Then, the maximized likelihood is a non-decreasing function of the number of states and can not be used as a model selection criterion.

In HMMs with Gaussian mixture emission densities, there is an additional non-identifiability issue. For example, consider an HMM with two states, such that one of the states has a two-component mixture emission density, and the other state has a single-component Gaussian emission density. It happens that this HMM is equivalent to another one with three states characterized by single-Gaussian emission densities. A formal proof of equivalence between an HMM with more than one Gaussian per state, and an HMM with more states but only one Gaussian per state, is presented in Appendix B. Supported by this equivalence, we will consider only HMMs with one Gaussian component per state, and focus only on the selection of the number of states (which we will denote as k).

To emphasize the model selection issue, in the sequel we will denote an HMM with k states as $\boldsymbol{\lambda}_k$.

3.1. Bayesian inference criterion

In the so-called BIC, the maximized likelihood is penalized by the model complexity, measured by the number of free parameters in $\boldsymbol{\lambda}_k$. Let $\mathcal{O}$ denote the observed data-set, and let n be the total number of observations in $\mathcal{O}$, i.e., $n = \sum_{l=1}^{L} T_l$. Under the BIC criterion, the optimal number of states is the maximizer of $\mathrm{BIC}(k)$, $\hat{k}_{\mathrm{BIC}} = \arg\max_k \mathrm{BIC}(k)$, where

$$\mathrm{BIC}(k) = \log p(\mathcal{O}|\hat{\boldsymbol{\lambda}}_k) - \frac{N_k}{2}\log(n). \tag{5}$$

In Eq. (5), $\hat{\boldsymbol{\lambda}}_k$ denotes the ML estimate of the model with k states, and N_k is the total number of free parameters of $\hat{\boldsymbol{\lambda}}_k$.

3.2. Mixture minimum description length for HMM

To explain the rationale behind MMDL, we start with the standard MDL criterion (Rissanen, 1986) which coincides with BIC (Eq. (5)). Notice that in the BIC/MDL criterion, each parameter has

equal weight in the penalty term, $\log(n)/2$. In the mixture of Gaussians case, MMDL is based on the following observation: the parameters of the jth component are actually estimated from the observations that were generated by that component, not from all the observed data. Moreover, the expected number of samples obtained from the jth component is nc_j, where c_j is the probability of the jth component. The MMDL criterion for mixtures is then obtained by penalizing each parameter of component j by $\log(nc_j)/2$ (instead of the standard $\log(n)/2$), considering the quantity nc_j can be seen as an "effective sample size" for the jth component.

A similar reasoning can be followed in the HMM context, but care must be taken in the definition of the "effective sample size", because here there is nothing similar to the component probability c_j. We start by decomposing N_k as $N_k^{\boldsymbol{A}} + N_k^{\boldsymbol{\pi}} + N_k^{\boldsymbol{B}}$, denoting the number of parameters of the transition matrix $\boldsymbol{A}$, of the initial state probability $\boldsymbol{\pi}$, and of the emission probability density function $\boldsymbol{B}$, respectively. Following the MMDL rationale, we will weight the emission probability parameters of each state using the "effective sample size" corresponding to that state. The elements of the transition matrix and of the initial state probability vector will be weighted with the standard $\log(n)/2$, since they are estimated from all the samples.

The role of "state probabilities" (equivalent to $c_1, \ldots, c_k$, in the mixture case) will be played by the stationary probability distribution $\boldsymbol{p}_\infty = [p_\infty(1), \ldots, p_\infty(k)]$ (see Appendix A for the details about the computation of this probability). This seems to be a natural choice, since $\boldsymbol{p}_\infty$ represents the "average" occupation of each state, after the Markov chain has achieved the stationary state. Therefore, for an HMM with k states, the MMDL cost function will be

$$\mathrm{MMDL}(k) = \log p(\mathcal{O}|\hat{\boldsymbol{\lambda}}_k) - \frac{N_k^{\boldsymbol{A}} + N_k^{\boldsymbol{\pi}}}{2}\log(n) - \frac{N_1^{\boldsymbol{B}}}{2}\sum_{m=1}^{k}\log(np_\infty(m)),$$

where $N_1^{\boldsymbol{B}}$ is the number of parameters of the emission density of an HMM with just one state. Finally, notice that $\boldsymbol{A}$ has $k(k-1)$ free parameters, $\boldsymbol{\pi}$ has $(k-1)$ free parameters, and $N_1^{\boldsymbol{B}} = d + d(d+1)/2$, if we assume a full covariance matrix for each component and d-dimensional observations. Accordingly, after dropping all terms that do not depend on k,

$$\mathrm{MMDL}(k) = \log p(\mathcal{O}|\hat{\boldsymbol{\lambda}}_k) - \frac{k^2}{2}\log(n) - \frac{d^2+3d}{4}\sum_{m=1}^{k}\log(np_\infty(m)). \quad (6)$$

Notice that $\boldsymbol{p}_\infty = [p_\infty(1), \ldots, p_\infty(k)]$ is a function of $\hat{\boldsymbol{\lambda}}_k$ via the estimate of the transition matrix (see Appendix A).

4. The sequential state pruning strategy

The strategy is summarized as follows:

(1) Choose some model selection criterion, such as BIC/MDL (Eq. (5)), or MMDL (Eq. (6)); set $k_{\min}$ and $k_{\max}$, which are the minimum and maximum number of states allowed.
(2) Initialize the HMM estimation algorithm with $k_{\max}$ states using some standard heuristic (e.g., randomly, or using clustering). Let us denote as $\boldsymbol{\lambda}_k^{\mathrm{I}}$ the initial model used in the training procedure for the HMM with k states.
(3) While $k \geqslant k_{\min}$, do:
   (a) run the Baum–Welch algorithm until some convergence criterion is met, let $\hat{\boldsymbol{\lambda}}_k$ be the set of estimated parameters;
   (b) compute and store the value of the model selection criterion; let this be denoted as C_k;
   (c) find the least probable state (i.e., the smallest element of $\boldsymbol{p}_\infty$);
   (d) prune the least probable state and deleting the corresponding elements from $\boldsymbol{A}$, $\boldsymbol{B}$, obtaining a reduced model $\bar{\boldsymbol{\lambda}}$;
   (e) set $\boldsymbol{\lambda}_{k-1}^{\mathrm{I}} \leftarrow \bar{\boldsymbol{\lambda}}$, and $k \leftarrow k-1$.
(4) The final chosen model, $\boldsymbol{\lambda}^*$, is the one yielding the maximum of the selection criterion. Formally:

$$\boldsymbol{\lambda}^* = \hat{\boldsymbol{\lambda}}_{k^*}, \quad \text{where} \quad k^* = \arg\max_k C_k.$$

The computational overhead introduced by this procedure is due mostly to the computation of $\boldsymbol{p}_\infty$,

involving the computation of eigenvalues of $\boldsymbol{A}$. However, this is computed only once for each k, at the end of the Baum–Welch training session. For the MMDL approach, there is actually no computational overhead, since $\boldsymbol{p}_\infty$ is also needed when evaluating the selection criterion (Eq. (6)).

5. Testing

To assess the performance of the proposed approach, we have performed tests in which we compare two strategies:

- Standard BIC (or MMDL) method: we train one HMM for each k (number of states), with k varying from $k_{\max}$ to $k_{\min}$. Each learning session (Baum–Welch algorithm) is initialized using a Gaussian mixture model, which is better than the usual random initilization. Each learning session is stopped when the relative increase of the likelihood function falls below a threshold. For each k, we compute and store the BIC (or MMDL) value, and, finally, we choose the model yielding the best value.
- Pruning BIC (or MMDL) method: as described in Section 4.

In all the HMMs considered in this paper, the emission probability density of each state is a single univariate Gaussian. The two strategies are compared in terms of (1) accuracy of the model size estimation, (2) total computational cost (total number of iterations) required by Baum–Welch procedure, and (3) classification accuracy on three recognition tasks (one synthetic and two real data problems).

5.1. Accuracy of model selection

We have tested our procedure on three different problems. For each one, the test set contains 5 sequences, each 400 observations long, synthetically generated from a known HMM. To increase statistical significance, all experiments were repeated 50 times. We set $k_{\min}$ and $k_{\max}$ to 2 and 10, respectively.

The first model is shown in Fig. 1(a): $\boldsymbol{A}$ is the transition matrix, $\boldsymbol{\pi}$ is the initial state probability, and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the means and variances of the Gaussian emission densities of each state. This is a relatively simple model, where Gaussians of different states are very well separated. The results (in Table 1(panel a)) show that, regarding accuracy in the selection of the true k, all model selection procedures perform perfectly. Regarding the computational requirements, the pruning strategy is less demanding, requiring about the 77% of the number of Baum–Welch iterations of the normal procedure. The second model is more challenging, since two of the emission Gaussians overlap: common mean but different variances (see Fig. 1(b)). Also in this case, there is no difference between BIC and MMDL, but there is a great difference between the two training strategies. In Table 1(panel b), the accuracies are reported, showing that the pruning methodology performs perfectly, with 100% accuracy, whereas the accuracy is 54% for the standard algorithm. Nearly one half of the models selected with the normal strategy have a wrong number of states (typically too many). This is confirmed in Fig. 2(a), where histograms of the selected numbers of states are shown. Also in this case, the average number of iterations required by the pruning method is significantly lower.

The third model is obtained from the second one by changing the transition matrix (see Fig. 1(c)). From Table 1(panel c) and Fig. 2(b), it is clear that, also in this example, the pruning strategy performs better, with a nearly perfect accuracy, versus about 86% for the standard method. The average number of iterations required by the pruning strategy is 52.7% of that required by the normal procedure.

5.2. Classification accuracy

We now study the performances of the proposed method in terms of classification accuracy on recognition tasks, using both synthetic and real data.

5.2.1. Synthetic data

In order to test the classification accuracy of the two methods, we have used the following testing procedure.

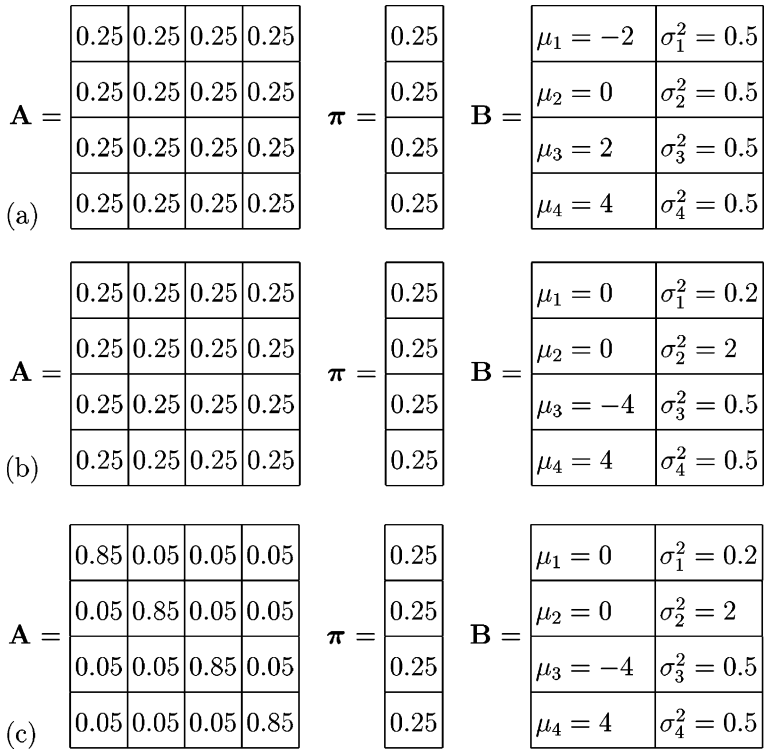

Fig. 1. Three models for the synthetic data test: A is the transition matrix, π is the initial state probability, μ and σ the parameters of the emission density.

Table 1
Results on synthetic data

	Selection accuracy	Average iterations
(a) First experiment		
Standard BIC	50/50 (100%)	110
Standard MMDL	50/50 (100%)	110
Pruning BIC	50/50 (100%)	84
Pruning MMDL	50/50 (100%)	84
(b) Second experiment		
Standard BIC	27/50 (54%)	175
Standard MMDL	27/50 (54%)	175
Pruning BIC	50/50 (100%)	103
Pruning MMDL	50/50 (100%)	103
(c) Third experiment		
Standard BIC	43/50 (86%)	186
Standard MMDL	43/50 (86%)	186
Pruning BIC	49/50 (98%)	98
Pruning MMDL	49/50 (98%)	98

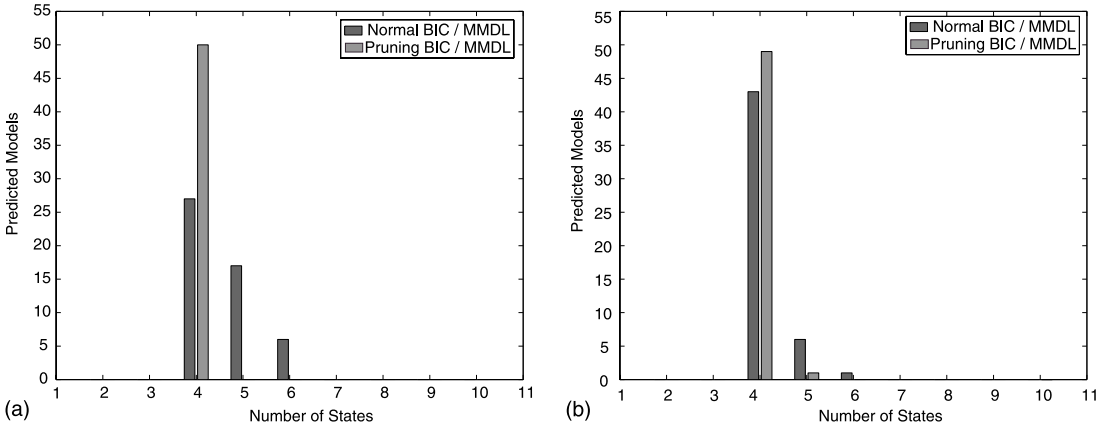

Fig. 2. Histograms of the selected number of states for the standard and the pruning strategies; the correct number of states is 4. (a) Second and (b) third experiments.

- Two training sets are generated, according to two models, each corresponding to one of two different classes.
- Two HMMs, one for each class, are trained using both methods (pruning and standard) and both model selection criteria (BIC and MMDL).
- Two test sets from the same true models are then generated.
- The classification accuracy using these test sets (a sequence is assigned to the class whose model has the highest likelihood), are finally estimated.

For each model, the training set contained 5 sequences of length 400. The test set was composed by 20 sequences, 10 from the first class and 10 from the second. To increase statistical significance, experiments were repeated 25 times. As before, $k_{\min} = 2$ and $k_{\max} = 10$.

In the first experiment, the HMM models used for each class are those shown in Fig. 1(b) and (c), only differing in the transition matrix $\boldsymbol{A}$. The experimental results are shown in Table 2(panel a). Both techniques perform perfectly, with the pruning method requiring fewer Baum–Welch iterations.

The second classification task considered was a very difficult one: the first model is the one shown in Fig. 1(b), the second one is almost the same, the only difference being the variance of the Gaussian of the first state: 0.4 instead of 0.2. The two HMMs are quite similar, but, as we can see in Table 2(panel b), the classification performance is very good. More in detail, the pruning strategy is better, with an accuracy of 98%, i.e., 6% above that of the standard procedure. In this case, the effectiveness of the learning is crucial for the correct discrimination. Moreover, the number of iterations required in the training phase is reduced for the pruning method, nearly 65% of the standard method.

5.2.2. Real data

Finally, we have conducted two classification experiments with real data. The first one involves a 2D shape recognition problem, using HMMs as described in (Bicego and Murino, submitted for publication). The second is a face recognition experiment, using HMMs as proposed in (Kohir and Desai, 1998).

The 2D shape recognition test is performed on the data set described in (Sebastian et al., 2001), which has four classes, each containing 12 different shapes. An object from each class is shown in Fig. 3. Just as in the synthetic experiments above reported, the pruning method performs better on this real-data problem (see Table 3), and involves a smaller computational burden. The classification accuracies reported are computed using the leave-one-out method, and experiments were repeated 10 times to increase the statistical significance.

Table 2
Classification accuracy on synthetic data

	Classification accuracy		Average iterations
	Mean	Standard deviation	
(a) First experiment			
Normal BIC	20/20 (100%)	0/20	110
Normal MMDL	20/20 (100%)	0/20	110
Pruning BIC	20/20 (100%)	0/20	84
Pruning MMDL	20/20 (100%)	0/20	84
(b) Second experiment			
Normal BIC	18.44/20 (92.2%)	2.31/20	163
Normal MMDL	18.44/20 (92.2%)	2.31/20	163
Pruning BIC	19.60/20 (98.0%)	0.76/20	107
Pruning MMDL	19.60/20 (98.0%)	0.76/20	107

Fig. 3. Examples of shapes from database used.

Face recognition was recently addressed using HMMs (e.g., Samaria, 1994; Achermann and Bunke, 1996; Kohir and Desai, 1998; Nefian and Hayes, 1998; Eickeler et al., 2000). In particular, techniques proposed in (Kohir and Desai, 1998) and (Eickeler et al., 2000) outperform all other methods available in the literature on a standard database, like ORL (Olivetti Research Ltd.). Here we use the method proposed in (Kohir and Desai, 1998), that considers discrete cosine transform (DCT) coefficients as features. Given a sequence of sub images of the face image, the DCT coefficients of each sub image are computed, and vectorized using a *zig-zag* scan. The number of coefficients chosen determines the dimensionality of the observation, and 10 coefficients are used in our experiment. The sequence of sub images is obtained by sliding over the face image a square fixed size window, in a raster scan fashion, with a predefined overlap. The window size and the overlap ratio were fixed respectively to 16% and 50%. The experiments have been conducted on the ORL database, [1] which consists in 40 subjects with 10 faces each. For each subject, five faces were used for training and the others for testing. The results, shown in Table 4, were obtained by repeating the experiments 25 times and averaging the results. Results are very satisfactory: the classification accuracies are similar, but our method reduces substantially the number of the iterations required.

A general consideration could be done looking at the standard deviations presented in all results tables: performances of the proposed approach are more stable, as the corresponding standard deviations are lower than those obtained with standard techniques. This confirms the fact that with our method the initialization is better addressed, resulting in a more stable and initialization-independent training process.

5.3. Comparison between BIC and MMDL criterions

In all synthetic experiments, the BIC and MMDL criteria chose the same topology, leading to the same model selection accuracy. Nevertheless, in the real-data case, the MMDL criterion slightly outperforms BIC in the resulting classification accuracy, showing that, as claimed in (Figueiredo et al., 1999), in some cases this criterion is better able to select a better model structure. In this paper, we have not focused on the comparison between BIC and MMDL for HMMS in more difficult situations; we will present such a comparison in a forthcoming paper.

[1] Downloadable from http://www.uk.research.att.com/facedatabase.html.

Table 3
Classification accuracy on real data considering 2D shape classification

	Classification accuracy		Average iterations
	Mean	Standard deviation	
Normal BIC	44.4/48 (92.5%)	1.26/48	94.1
Normal MMDL	45.3/48 (94.37%)	0.95/48	94.1
Pruning BIC	45.7/48 (95.21%)	0.48/48	76.6
Pruning MMDL	45.7/48 (95.21%)	0.67/48	76.6

Table 4
Classification accuracy on real data: face recognition

	Classification accuracy		Average iterations
	Mean	Standard deviation	
Normal BIC	195/200 (97.5%)	1.54/200	86.2
Normal MMDL	195/200 (97.5%)	1.54/200	86.2
Pruning BIC	195.26/200 (97.63%)	0.95/200	51.4
Pruning MMDL	195.26/200 (97.63%)	0.95/200	51.4

6. Conclusions

In this paper, a new approach to the optimal selection of the structure of a HMM is proposed. The key idea is to perform a decreasing learning strategy, starting each training session from a "nearly good" configuration, derived from previous training by pruning the "least probable" state. The proposed strategy can be applied for all types of HMMs and can be used with any model selection criterion. In this work, we have considered the BIC, and we have adapted the MMDL criterion to the HMM case. Experimental results show that our approach is more accurate in finding the true model, is more effective in classification accuracy, while having reduced computational requirements. Moreover, the performances of the proposed approach are more stable, as the corresponding standard deviations are lower than those obtained with standard techniques. This confirms the fact that with our method the initialization is better addressed, resulting in a more stable and initialization-independent training process.

Appendix A. Stationary probability distribution P_∞

Consider the Markov chain $\boldsymbol{Q} = Q_1, Q_2, Q_3, \ldots$ with state set $S = \{S_1, \ldots, S_k\}$, stochastic transition matrix $\boldsymbol{A}$, and initial state probability $\boldsymbol{\pi}$. We can define the vector of state probabilities at time t as

$$\begin{aligned}\boldsymbol{p}_t &= [p_t(1), \ldots, p_t(j), \ldots, p_t(k)] \\ &= [P(Q_t = S_1), P(Q_t = S_2), \ldots, P(Q_t = S_k)].\end{aligned}$$

Of course, $\boldsymbol{p}_t$ can be computed recursively from $\boldsymbol{p}_1 = \boldsymbol{\pi}\boldsymbol{A}$, $\boldsymbol{p}_2 = \boldsymbol{p}_1\boldsymbol{A} = \boldsymbol{\pi}\boldsymbol{A}\boldsymbol{A}$, and so on. That is $\boldsymbol{p}_t = \boldsymbol{\pi}\boldsymbol{A}^t$.

We are interested in $\boldsymbol{p}_\infty$, which characterizes the equilibrium behavior of the Markov chain, i.e., when we let it evolve indefinitely. Since it is a stationary distribution, $\boldsymbol{p}_\infty$ has to be a solution of $\boldsymbol{p}_\infty = \boldsymbol{p}_\infty\boldsymbol{A}$, or, in other words, it has to be a left eigenvector of $\boldsymbol{A}$ associated with the unit eigenvalue. Under some conditions (see, e.g., Brémaud (1999), for details), the Perron–Frobenius theorem states that matrix $\boldsymbol{A}$ has a unit (left) eigenvalue and the corresponding left eigenvector is $\boldsymbol{p}_\infty$. All other eigenvalues of $\boldsymbol{A}$ are strictly less than 1, in absolute value. Finding $\boldsymbol{p}_\infty$ for a given $\boldsymbol{A}$ then amounts to solving the corresponding eigenvalue/eigenvector problem.

Appendix B. Equivalence between Gaussian HMMs

In this appendix, we show that, given an HMM $\boldsymbol{\lambda}$ with k states, where the emission probability of each state S_i is a mixture of (univariate or multi-

variate) Gaussians, each Gaussian having parameters θ_{im},

$$b(o|S_i) = \sum_{m=1}^{M_i} c_{im}\mathcal{N}(o|\theta_{im}), \tag{B.1}$$

then there is another HMM $\boldsymbol{\lambda}'$ with $k' = \sum_{i=1}^{k} M_i$ states, with only one Gaussian for state, that is equivalent to $\boldsymbol{\lambda}$. Here, equivalence is understood in a likelihood sense, that is, $P(\boldsymbol{o}|\boldsymbol{\lambda}) = P(\boldsymbol{o}|\boldsymbol{\lambda}')$, for any sequence $\boldsymbol{o} = o_1, o_2, \ldots, o_T$.

First we will describe how model $\boldsymbol{\lambda}'$ is built; subsequently we will show that the two models are equivalent. Given $\boldsymbol{\lambda} = (S, \boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{B})$, the equivalent model $\boldsymbol{\lambda}' = (S', \boldsymbol{A}', \boldsymbol{\pi}', \boldsymbol{B}')$ is defined as follows:

- *New states*: we split each state S_i into M_i states, one for each of the M_i Gaussians of the mixture of S_i. Thus we obtain $k' = \sum_{i=1}^{k} M_i$ states and

$$\begin{aligned} S' &= \{S'_1, \ldots, S'_{k'}\} \\ &= \{S'_{11}, \ldots, S'_{1M_1}, S'_{21}, \ldots, S'_{2M_2}, S'_{31}, \ldots, S'_{kM_k}\}, \end{aligned} \tag{B.2}$$

  where we have introduced the double index notation in which S'_{im} corresponds to the mth Gaussian of the original state S_i.
- *Emission probabilities*: naturally, the emission probability of state S'_{im} is the corresponding Gaussian

$$b'(o|S'_{im}) = \mathcal{N}(o|\theta_{im}). \tag{B.3}$$

- *State transition probability*: using the double index notation, where

$$A'_{ik,jm} = P(Q_{t+1} = S'_{jm}|Q_t = S'_{ik}), \tag{B.4}$$

  denotes the probability of going from state S'_{ik} to state S'_{jm}, we set

$$A'_{ik,jm} = A_{ij}c_{jm}, \tag{B.5}$$

  where c_{jm} is the mixing weight of the mth component from the original state S_j. Notice that $A'_{ik,jm}$ does not depend on k and that, as required,

$$\sum_{jm} A'_{ik,jm} = \sum_{j=1}^{k}\sum_{m=1}^{M_j} A'_{ik,jm} = \sum_{j=1}^{k} A_{ij}\sum_{m=1}^{M_j} c_{jm} = 1.$$

- *Initial state probability*: similarly to the previous definition, we set

$$\pi'(S'_{jm}) = \pi(S_j)c_{jm} \tag{B.6}$$

which is also clearly normalized.

The proof of the equivalence between the two HMMs uses the *forward–backward* procedure (see, e.g., Rabiner, 1989), the standard technique for computing $P(\boldsymbol{o}|\boldsymbol{\lambda})$. This technique is based on the *forward* variables $\alpha_t(S_i)$, defined as

$$\alpha_t(S_i) = P(o_1, \ldots, o_t, q_t = S_i|\boldsymbol{\lambda}), \tag{B.7}$$

which are iteratively computed according to

$$\alpha_1(S_i) = \pi(S_i)b(o_1|S_i), \tag{B.8}$$

$$\alpha_{t+1}(S_i) = b(o_{t+1}|S_i)\sum_{j=1}^{k}\alpha_t(S_j)A_{ji}. \tag{B.9}$$

Given the sequence $\boldsymbol{o} = o_1, \ldots, o_T$, $P(\boldsymbol{o}|\boldsymbol{\lambda})$ is computed by marginalization,

$$P(\boldsymbol{o}|\boldsymbol{\lambda}) = \sum_{i=1}^{k} P(o_1, \ldots, o_T, Q_T = S_i|\boldsymbol{\lambda}) = \sum_{i=1}^{k}\alpha_T(S_i). \tag{B.10}$$

With the goal of showing that $P(\boldsymbol{o}|\boldsymbol{\lambda}) = P(\boldsymbol{o}|\boldsymbol{\lambda}')$, let us rewrite $P(\boldsymbol{o}|\boldsymbol{\lambda}')$ as

$$P(\boldsymbol{o}|\boldsymbol{\lambda}') = \sum_{i=1}^{k'}\alpha_T(S'_i) = \sum_{i=1}^{k}\sum_{m=1}^{M_i}\alpha_T(S'_{im}), \tag{B.11}$$

that is, using the double index notation introduced in (B.2). Let us also define

$$\alpha'_T(S_i) = \sum_{m=1}^{M_i}\alpha_T(S'_{im}). \tag{B.12}$$

Clearly, if we show that, for $i = 1, \ldots, k$,

$$\alpha_T(S_i) = \alpha'_T(S_i) \tag{B.13}$$

then, we will be able to conclude, as desired, that

$$\underbrace{\sum_{i=1}^{k}\alpha_T(S_i)}_{P(\boldsymbol{o}|\boldsymbol{\lambda})} = \sum_{i=1}^{k}\alpha'_T(S_i) = \sum_{i=1}^{k}\sum_{m=1}^{M_i}\alpha_T(S'_{im}) = \underbrace{\sum_{i=1}^{k'}\alpha_T(S'_i)}_{P(\boldsymbol{o}|\boldsymbol{\lambda}')}.$$

We will now show (B.13) by induction on the length T of the sequence $\boldsymbol{o}$.

- We start with $T = 1$. From (B.8), we know that

$$\alpha_1(S_i) = \pi(S_i)b(o_1|S_i) = \pi(S_i)\sum_{m=1}^{M_i} c_{im}\mathcal{N}(o_1|\theta_{im}). \quad \text{(B.14)}$$

Now, we can also write

$$\alpha'_1(S_i) = \sum_{m=1}^{M_i}\alpha_1(S'_{im}) = \sum_{m=1}^{M_i}\pi'(S'_{im})\mathcal{N}(o_1|\theta_{im}) = \sum_{m=1}^{M_i}\pi(S_i)c_{im}\mathcal{N}(o_1|\theta_{im}) = \pi(S_i)\sum_{m=1}^{M_i} c_{im}\mathcal{N}(o_1|\theta_{im}),$$

where the first equality is (B.12), the second one is (B.8), the third results from the definitions of the model $\boldsymbol{\lambda}'$ (B.3) and (B.6). Then we have shown that $\alpha_1(S_i) \equiv \alpha'_1(S_i)$.

- To show the recursion, we have to prove that

$$\alpha_T(S_i) = \alpha'_T(S_i) \Rightarrow \alpha_{T+1}(S_i) = \alpha'_{T+1}(S_i). \quad \text{(B.15)}$$

Invoking (B.9), we can write

$$\alpha_{T+1}(S_i) = \left[\sum_{j=1}^{k}\alpha_T(S_j)A_{ji}\right]\left(\sum_{m=1}^{M_i} c_{im}\mathcal{N}(o_{T+1}|\theta_{im})\right). \quad \text{(B.16)}$$

Also, by using (B.12), and again (B.9), we have

$$\begin{aligned}\alpha'_{T+1}(S_i) &= \sum_{m=1}^{M_i}\alpha_{T+1}(S'_{im}) \\ &= \sum_{m=1}^{M_i}\sum_{j=1}^{k}\sum_{\ell=1}^{M_\ell}\alpha_T(S'_{j\ell})A_{j\ell,im}\mathcal{N}(o_{T+1}|\theta_{im}) \\ &= \sum_{m=1}^{M_i}\sum_{j=1}^{k}\sum_{\ell=1}^{M_\ell}\alpha_T(S'_{j\ell})c_{im}A_{ji}\mathcal{N}(o_{T+1}|\theta_{im}) \\ &= \sum_{m=1}^{M_i} c_{im}\mathcal{N}(o_{T+1}|\theta_{im})\sum_{j=1}^{k}A_{ji}\sum_{\ell=1}^{M_\ell}\alpha_T(S'_{j\ell}) \\ &= \left(\sum_{m=1}^{M_i} c_{im}\mathcal{N}(o_{T+1}|\theta_{im})\right)\sum_{j=1}^{k}A_{ji}\alpha'_T(S_j), \end{aligned} \quad \text{(B.17)}$$

where the third equality results from (B.5), and the last one from (B.12). Finally, comparing (B.17) with (B.16) clearly shows that the implication in (B.15) is true.

This concludes our proof that $P(\boldsymbol{o}|\boldsymbol{\lambda}) = P(\boldsymbol{o}|\boldsymbol{\lambda}')$.

References

Achermann, B., Bunke, H., 1996. Combination of face classifiers for person identification. In: Internat. Conf. on Pattern Recognition, pp. C416–C420.

Baum, L., 1970. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequality 3, 1–8.

Baum, L., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals Math. Statist. 41 (1), 164–171.

Bicego, M., Murino, V., submitted for publication. Investigating Hidden Markov Models' capabilities in 2D shape classification. IEEE Trans. Pattern Anal. Machine Intell.

Bicego, M., Dovier, A., Murino, V., 2001. Designing the minimal structure of hidden Markov models by bisimulation. In: Figueiredo, M., Zerubia, J., Jain, A. (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer, pp. 75–90.

Bishop, C.M., 1995. Neural Network for Pattern Recognition. Clarendon Press.

Brand, M., 1999. An entropic estimator for structure discovery. In: Kearns, M., Solla, S., Cohn, D. (Eds.), Advances in Neural Information Processing Systems, vol. 11. MIT Press Cambridge, MA.

Brémaud, P., 1999. Markov Chains. Springer-Verlag.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. B 39, 1–38.

Eickeler, S., Kosmala, A., Rigoll, G. 1998. Hidden Markov Model based continuous online gesture recognition. In: IEEE Proc. Internat. Conf. on Pattern Recognition. vol. 2, pp. 1206–1208.

Eickeler, S., Müller, S., Rigoll, G., 2000. Recognition of JPEG compressed face images based on statistical methods. Image Vision Comput. 18, 279–287.

Figueiredo, M., Leitao, J., Jain, A., 1999. On fitting mixture models. In: Hancock, E., Pellilo, M. (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer Verlag, pp. 54–69.

Forney, G., 1973. The Viterbi algorithm. Proc. IEEE 61, 268–278.

Hu, J., Brown, M., Turin, W., 1996. HMM based online handwriting recognition. IEEE Trans. Pattern Anal. Machine Intell. 18 (10), 1039–1045.

Hughey, R., Krogh, A., 1996. Hidden Markov Model for sequence analysis: Extension and analysis of the basic method. Comput. Appl. Biosci. 12, 95–107.

Jebara, T., Pentland, A., 1999. Action reaction learning: Automatic visual analysis and synthesis of interactive behavior. In: Proc. Internat. Conf. on Comput. Vision Systems.

Juang, B., Levinson, S., Sondhi, M., 1986. Maximum likelihood estimation for multivariate mixture observations of Markov chain. IEEE Trans. Inform. Theory 32 (2), 307–309.

Kohir, V.V., Desai, U.B., 1998. Face recognition using dct-hmm approach. In: Workshop on Advances in Facial Image Analysis and Recognition Technology (AFIART), Freiburg, Germany.

Li, D., Biem, A., Subrahmonia, J., 2001. HMM topology optimization for handwriting recognition. In: Proc. of IEEE Interna. Conf. on Acoust., Speech, and Signal Process, vol. 3. pp. 1521–1524.

Nefian, A.V., Hayes, M.H., 1998. Hidden Markov models for face recognition. In: Proc. IEEE Internat. Conf. on Acoust., Speech and Signal Process. (ICASSP), Seattle, pp. 2721–2724.

Oliver, J., Baxter, R., Wallace, C., 1996. Unsupervised learning using MML in machine learning. In: Proc. of 13th Internat. Conf. (ICML 96). Morgan Kaufmann Publishers, pp. 364–372.

Rabiner, L., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of IEEE 77 (2), 257–286.

Raftery, A., 1995. Bayesian model selection in social research. Sociol. Methodol., 111–196.

Rissanen, J., 1986. Stochastic complexity and modeling. The Annals Statist., 14.

Samaria, F., 1994. Face recognition using hidden markov models. Technical report, Ph.D. thesis, Engineering Department, Cambridge University.

Schwarz, G., 1978. Estimating the dimension of a model. Ann. Statist. 6 (2), 461–464.

Sebastian, T., Klein, P., Kimia, B., 2001. Recognition of shapes by editing shock graphs. In: Proc. Internat. Conf. on Computer Vision, pp. 755–762.

Stolcke, A., Omohundro, S., 1993. Hidden Markov model induction by Bayesian model merging. In: Hanson, S., Cowan, J., Giles, C. (Eds.), Advances in Neural Information Processing Systems, 5. Morgan Kaufmann, San Mateo, CA, pp. 11–18.

Stone, M., 1974. Cross-validatory choice and assessment of statistical predictions. J. Roy. Statist. Soc. B 36, 111–147.

Zimmermann, M., Bunke, H., 2001. Hidden Markov model length optimization for handwriting recognition systems. TR IAM-01-003 University of Bern.