PERGAMON

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

Pattern Recognition 37 (2004) 1325–1337

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Diagnostic tools for evaluating and updating hidden Markov models

Brendan McCane[a,*], Terry Caelli[b]

[a] *Department of Computer Science, University of Otago, Dunedin, New Zealand*
[b] *Department of Computer Science, University of Alberta, Edmonton, Canada*

Received 14 August 2003; received in revised form 17 December 2003; accepted 17 December 2003

Abstract

In this paper we consider two related problems in hidden Markov models (HMMs). One, how the various parameters of an HMM actually contribute to predictions of state sequences and spatio-temporal pattern recognition. Two, how the HMM parameters (and associated HMM topology) can be updated to improve performance. These issues are examined in the context of four different experimental settings from pure simulations to observed data. Results clearly demonstrate the benefits of applying some critical tests on the model parameters before using it as a predictor or spatio-temporal pattern recognition technique.

Keywords: Hidden Markov models

1. Introduction

Hidden Markov models (HMMs) have become a standard method for encoding, recognizing and predicting sequential patterns of univariate and multivariate observation data. The Viterbi algorithm has been the most popular method for predicting optimal state sequences and it's associated maximum posterior probability (log(MAP)) score is typically used for temporal pattern recognition and classification. Similarly, the Baum Welch algorithm, a form of expectation-maximization (EM), and its variations, has been the predominant model estimation technique, given a model topology [1,2].

A number of methods for generating and updating HMM topologies (the number of states and their state transitions) have also been explored in recent years. These methods include state splitting and deletions [3] using MDL and entropy MAP-based methods [4] where the aim is typically to maximize predictions with the smallest number of model parameters, θ, given an observation sequence. Other work on topology estimation has focused on comparing HMM model performance. For example, Lyngso et al. [5] focus on comparing HMMs in terms of the co-emission probability of state emissions. Bahlman et al. [6] use Bayesian estimates of HMM state correspondences. Balasubramanian [7] has performed extensive theoretical work on finding equivalent HMMs based on the equal probability of the observation sequences alone, and regardless of the number of internal states. He then uses this result to define conditions and an algorithm for finding minimal Generalized Markov Models (an HMM with the parameter positivity constraint relaxed). In some sense these minimal models are optimal as they contain the fewest number of parameters for the same result —given an observation sequence.

The problem is, however, the observation sequences, $\vec{O}$. For example, how do we know they are representative, unbiased or sufficient to base model estimation and update upon? Consequently in this paper, we first explore what can be concluded about a HMM's model parameters without considering any particular observation sequence. Further, even if the estimation method is based upon observations it is still important to interpret

* Corresponding author.
E-mail addresses: mccane@cs.otago.ac.nz (B. McCane), tcaelli@cs.ualberta.ca (T. Caelli).

0031-3203/$30.00
doi:10.1016/j.patcog.2003.12.017

the roles of the model parameters in performance of the model.

First, some definitions. We follow the HMM nomenclature of Rabiner [1]. The discrete HMM model, λ, consists of three components $\lambda = \{A, B, \pi\}$ having N states and M distinct observation symbols; where $A = \{a_{ij}\}$ is an $N \times N$ state transition probability matrix and

$$a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i], \quad 1 \leqslant i, \ j \leqslant N, \tag{1}$$

where q corresponds to the state random variable (*r.v.*), $B = \{b_j(k)\}$ is an $N \times M$ matrix which is the probability distribution of observation symbol, o, given state j, where

$$b_j(k) = P[o = k \mid q = S_j], \quad 1 \leqslant j \leqslant N, \ 1 \leqslant k \leqslant M, \tag{2}$$

for the observation *r.v.*, o; and $\pi = \{\pi_i\}$ is either the initial state distribution where

$$\pi_i = P[q_1 = S_i], \quad 1 \leqslant i \leqslant N \tag{3}$$

or the steady state distribution:

$$\forall t \geqslant t^* \quad \pi_i = P[q_t = S_i], \quad 1 \leqslant i \leqslant N. \tag{4}$$

In the HMM literature, the former interpretation is common, and the latter is common in the Markov chain literature.

2. Some initial observations

Critical to model estimation are the forward and backward operators. For each state, S_j, at time, $t + 1$, we have the recursive forms

$$\alpha_j(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_j(o_{t+1}), \quad 1 < j \leqslant N \tag{5}$$

and

$$\beta_i(t) = \sum_{j=1}^{N} \beta_i(t+1) a_{ij} b_j(o_{t+1}), \quad 1 \leqslant i < N, \tag{6}$$

respectively. For model prediction (using the Viterbi algorithm) the forward operator is transformed into

$$\phi_j(t+1) = \max_i \{\phi_i(t) a_{ij} b_j(o_{t+1})\}. \tag{7}$$

In all three cases, the product term $a_{ij} b_j(o_{t+1})$ plays a key role and the matrix

$$C = A^{\mathrm{T}} B \quad \Leftrightarrow \quad c_{ik} = \sum_{j=1}^{N} a_{ij} b_i(k) \tag{8}$$

encodes all terms in these equations. Balasubramanian [7] also noted the importance of $A^{\mathrm{T}}B$ in defining model equivalence and minimal models.

Further, if the A matrix has unit rank then we know that the emission pdfs are identical to the steady-state (invariant) distribution of the underlying Markov chain [8]. This follows from the fact that in such situations the first left eigenvector is identical to the common row vector. In this case, then

$$\sum_{j=1}^{N} a_{ij} b_j(k) = \sum_{j=1}^{N} a_j b_j(k) = \sum_{j=1}^{N} \pi_j b_j(k). \tag{9}$$

Also, for estimation, then, we obtain

$$\phi_j(t) = \max_j \{\phi_i(t-1) \pi_j b_{jk}(t)\} \tag{10}$$

$$= \max_j \{p(o(t)/S_j) p(S_j)\}. \tag{11}$$

That is, the computation at each event corresponds to that of a MAP Bayesian classifier as the Markov constraint does not differentiate values of $\phi_j(t)$. A similar situation applies to the B matrix. That is, a unit rank B matrix adds no new information to predictions from observations. Consequently, it is important to assess the contributions of these two major sources of information in model interpretation, estimation and prediction.

For example, consider the following three A and B matrices (2 state and 2 observation symbol cases):

$$A_1 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.8 & 0.2 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$$

and

$$B_1 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.8 & 0.2 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}.$$

These matrices can generate nine different models that could well have been estimated from observation sequences but illustrate quite important differences in what we mean by an "HMM" model. Since it is also well known that the L_1 norm of the eigenvector corresponding to the unit left-hand eigenvector of A corresponds to the invariant (steady state or initial) distribution of a Markov chain [8], then there is no particular need to examine the π vector in this analysis. So, consider a model with:

- A_1, B_1. Although it may be valid, it would correspond to random observation and inferred state sequences and no predictive power.
- One with A_1, B_2 would fail in prediction of the optimal state sequence due to the correlation between the evidence for observations given states and, again, the lack of evidence from previous states values.
- On the other hand, one with A_1, B_3 would provide quite adequate identification of the optimal state sequence based purely on the current evidence at any time step—the B matrix being used as a Bayesian (ML or MAP(using π)) classifier.

- In similar ways, we can observe just how the independence and uncertainty of evidence from A_2, A_3 would contribute evidence to inferring the optimal state sequence. For example, evidence for a particular state from the previous one using A_2 is totally ambiguous.

The above observations and examples indicate the importance of the model parameters in determining performance and discriminating between models even without considering any particular observation sequence. Although these observations are not conducive to, by definition, MAP analysis, other objective techniques can be used to determine two key HMM components of a given HMM: its *ambiguity* and *uncertainty*. Here we explore the use of the Inverse Condition Number and Residual Sum Vectors to identify model parameters and updates and, when observations are available, we explore the Conditional Information of the model components to identify their contributions to optimal performance.

3. The augmented matrix

Following on from these initial observations, our aim is to develop a method for assessing the contributions of the HMM components (specifically, the A and B matrices) to model performance. Although, as already discussed, the product matrix, $A^{\mathrm{T}}B$, plays a critical role in recognition and prediction, in order to enable model updating we explore characteristics of the more informative row augmented matrix:

$$C = A \mid B. \tag{12}$$

In the light of the above discussion, it is clear that the more linearly independent the rows of C are, the derivation of optimal state sequences and identification of temporal patterns becomes more unique for the model. This leads to the following measures that provide objective ways of determining just how the model parameters can contribute to the prediction of optimal state sequences—for any observation sequence.

3.1. HMM component analysis and the inverse condition number

In practice, HMMs are typically focused on predicting the optimal (MAP) state sequences from observations. However, some properties of these states can already be deduced from the model components, per se. First, the Markov chain component. For a Markov process, a well-known way of defining the distance between the current state transition probabilities and the steady state density function (invariant pdf) is from the total of the left-handed "residual eigenvalues" of A ($\mu_{res}(A)$)—the total of all the sub-dominant eigenvalues [8]. When all rows of the Markov chain are identical the Markov chain condition breaks down in so far as

$$P[\vec{S}_{t+1} \mid \vec{S}_t] = P[\vec{S}_{t+1}] = P[\vec{S}_t] = \pi. \tag{13}$$

Consequently, $\mu_{res}(A) = \sum_{i=2}^{N} \mu_i^2(A)$ provides a measure of how ergodic the process is and consequently the potential for generating variable state sequences as a function of the observation.

Secondly, as the rows of the state-dependent observation matrix, B, become more correlated, the evidence for a specific state from observations decreases. Similar to the A matrix steady-state condition, observations do not evidence any state when the B matrix has only one non-zero eigenvalue, leading to

$$P[O_k \mid S_i] = P[O_k]. \tag{14}$$

In all then, using the singular values of C, σ, the inverse condition number (ICN) of C [9] is

$$\gamma^{-1} = \sigma_{min}/\sigma_{max}, \tag{15}$$

where σ_{max} is the largest singular value of C and σ_{min} is the smallest, is an appropriate normalized measure of the "HMM bandwidth" in so far as $\gamma^{-1} = 1.0$ indicates that all states can be realized within the limits of the steady-state probabilities and state-dependent observations. However, $\gamma^{-1} = 0$ results in a "zero-bandwidth" HMM in so far as the process, on any experiment, does not provide any predictive information about state sequences except those provided by the prior or steady-state conditions.

3.2. HMM residuals

Such measures indicate how well the HMM is likely to cover the full dimensionality of the model parameter space. However, it does not measure where rank deficiencies may reside. To measure this, we let $C_{\setminus i}$ be the matrix C with row i (r_i), removed. We then compute $P_{C_{\setminus i}}(r_i)$, the projection of r_i onto the span of the vectors defined by the row space of $C_{\setminus i}$. The residual vector

$$e_i = r_i - P_{C_{\setminus i}}(r_i), \tag{16}$$

encodes the degree of redundancy of the complete description of state i (row i) as it contains both state transition and observation dependencies. The complete residual matrix

$$E = [\vec{e}_1 \vec{e}_2, \ldots, \vec{e}_N]^{\mathrm{T}} \tag{17}$$

defines the distribution of parameter (both state and observation) dependencies/redundancies and it can be used to identify exactly which states (and symbols) are problematic. If a particular element is close to 0, then the corresponding HMM element is linearly dependent on other rows in the matrix. If the element is close to the original element, then it is linearly independent of the other rows and therefore an important element. If a whole row is close to 0, then it indicates that the corresponding state is redundant and could be removed from the HMM.

Despite the usefulness of the residual matrix, it can be demanding to interpret. Therefore we use a simpler measure based on the residual matrix:

$$s_{\cdot j} = \sqrt{\sum_i e_{ij}^2}, \tag{18}$$

for each column of E, where e_{ij} denotes the element in row i and column j. The maximum value of any element e_{ij} is 1.0. Now, if the row space of each of the C_i's is orthogonal, then the maximum value of $s_{\cdot j}$ would also be 1.0. However, since the row space is typically not orthogonal, $s_{\cdot j}$ can be larger than 1.0. Nevertheless, a value of $s_{\cdot j}$ near 1.0 indicates that symbol j is quite independent of other symbols and is therefore important for the HMM. On the other hand, if $s_{\cdot j}$ is close to 0, there are two possibilities. Either state or symbol j is highly dependent on the other symbols and is therefore not particularly useful to the HMM, or it is unlikely to occur. In the former case, symbol j can be safely removed from the HMM.

The measure

$$s_{i\cdot} = \sqrt{\sum_j e_{ij}^2}, \tag{19}$$

defines the degree to which a given state is redundant over both states and symbols. As with $s_{\cdot j}$, increases in $s_{i\cdot}$ indicate the independence of a given state with respect to both the Markov condition and the state-dependent observations.

3.3. The conditional information of an HMM

We now consider how the A and B parameters contribute to the prediction of state sequences given a model *and* observations. As already mentioned, this has been the standard method for model evaluation and update. Here we show how conditional information [10] can be used to assess the contributions of these different components.

If the B matrix is unambiguous (for example, orthogonal with an ICN of 1.0) then a direct use of either maximum likelihood (ML: $\max_S \{P[O(t) \mid S]\}$) or maximum posterior probability (MAP: $\max_S \{P[S \mid O(t)]=P[O(t) \mid S]P[S]\}$ would suffice to predict the most likely state at time, t. This condition would eliminate the need for the Markov component (A matrix) by use of a simple Bayesian (ML or MAP) classifier. Conversely, if the model B matrix is ambiguous (ICN of 0), we may as well dispense with the observation part and simply use the Markov component of the HMM to determine the most likely state sequence given the Markov model. Accordingly, we show how Conditional Information can be used to tease out the contributions of each component to the solutions for optimal state sequences.

We have investigated this measure using the following procedure. Given a model and an input observation sequence we generate two state sequences, one using the Viterbi algorithm with the entire HMM, $\vec{S}_v$, and the other with a Bayesian classifier using only the B matrix (ML classifier) resulting in $\vec{S}_b$. This latter condition assumes that the predictions at each time period are independent of all others —a condition consistent with the independence of observations over time for regular HMMs. Given the resultant two state sequences, $\vec{S}_v$ and $\vec{S}_b$, respectively, we can calculate the following quantities:

$$H(v \mid b) = H(v, b) - H(b), \tag{20}$$

where

$$H(v, b) = -\sum_{i,j} (P(S_v = i, S_b = j) \times \log P(S_v = i, S_b = j)) \tag{21}$$

and

$$H(b) = -\sum_j (P(S_b = j) \log P(S_b = j)). \tag{22}$$

$H(v \mid b)$ is the conditional entropy, and $P(S_v = i,\ S_b = j)$ is computed from the joint frequencies of the two state sequences. This measures the amount of information about the Viterbi solution given the Bayesian classifier solution. The residual information

$$R(v \mid b) = H(v) - H(v \mid b) \tag{23}$$

provides a measure of how much information the A matrix, and the associated Viterbi algorithm, add to the complete optimal state sequence prediction.

In all, then, these measures offer clear ways for interpreting HMM performance and even the limits on the prediction of performance on any data set. In the following, we illustrate how these measures can be used to diagnose and even improve the performance of HMMs.

4. Experimental investigations

Since most applications of HMMs in pattern recognition are concerned with classification or identification of spatio-temporal patterns the typical criterion used is the Viterbi score defined as the log(MAP) probability of the optimal state sequence given an observation sequence and the model. We will show how this is less than an optimal method for discriminating between models and data as a function of the model's uncertainty. Further, the MAP value, per se, does not capture how well the HMM can predict or discriminate the state or observation sequences when this is a critical component to encoding and recognition. For this reason, we use a more stringent criterion to evaluate HMM performance: the degrees to which predicted optimal state sequences agree between different HMMs on the same data and the same HMM on different observation sequences. We consider four types of data: statistical experiments on binary state models, simulations representative of many gesture recognition tasks, some experimental data on

gesture recognition, and finally, a simple speech recognition task.

4.1. Simulation experiments

First, we have analyzed a very simple class of HMMs having two states and two observation symbols with systematically varying probabilities. From these HMMs, we generated test sequences using Monte Carlo sampling from which we could perform the proposed measurements. We generated a range of HMMs by varying both the A and B matrices independently from deterministic to random, with each of the A and B matrices being:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0.05 & 0.95 \\ 0.95 & 0.05 \end{bmatrix} \cdots \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \tag{24}$$

resulting in a total of 121 (11×11) HMMs.

In the first experiment, we tested how well we can consistently estimate state sequences generated from each of the above HMMs in the presence of "noisy" observation sequences. For each HMM two observation sequences of length 1000 were generated using Monte Carlo sampling of the model parameters. One of these observation sequences was then perturbed by randomly (uniformly) interchanging x% of the observations of a given sequence (noise permutations). The other sequence was not permuted for comparison purposes.

Optimal state sequences were generated using the Viterbi algorithm on the original observation sequence (without noise permutations) and each of the noise permuted ones. These predicted state sequences were then compared in terms of the percentage of identical versus different predicted states. This method provides a way of determining the degree to which the model can generalize to different observation sequences in predicting the optimal state sequence. The results are shown in Figure 1 where both axes vary from deterministic (=1) to random (=10). As can be seen from the figure, as both the A and B matrices become more random, the ability to estimate the optimal state sequence, for the original HMM, degrades. As expected, 50% correct (PC) indicates that the HMM performs no better than a random guess. Although the occurrence of noise degrades the performance of the HMM, it degrades gracefully.

Fig. 2 shows how the ICN correlates with performance (Pearson's $r = .8$) and the figure shows the least-squares regression line with the 95% confidence interval. This demonstrates how the ICN is a reasonable estimator of HMM performance when dealing with the normal uses of HMMs when there is a need to accommodate generalizations of the model: to apply when data is not exactly consistent with it in varying degrees. Although the ICN defines ambiguity it does not separate it from uncertainty. For example, the deterministic two-state Markov matrix with $(1, 0)$ rows has a zero ICN as does one with $(.5, .5)$ rows. However, besides these extreme cases both terms are typically correlated explaining the result shown simply for ICN in Fig. 2.

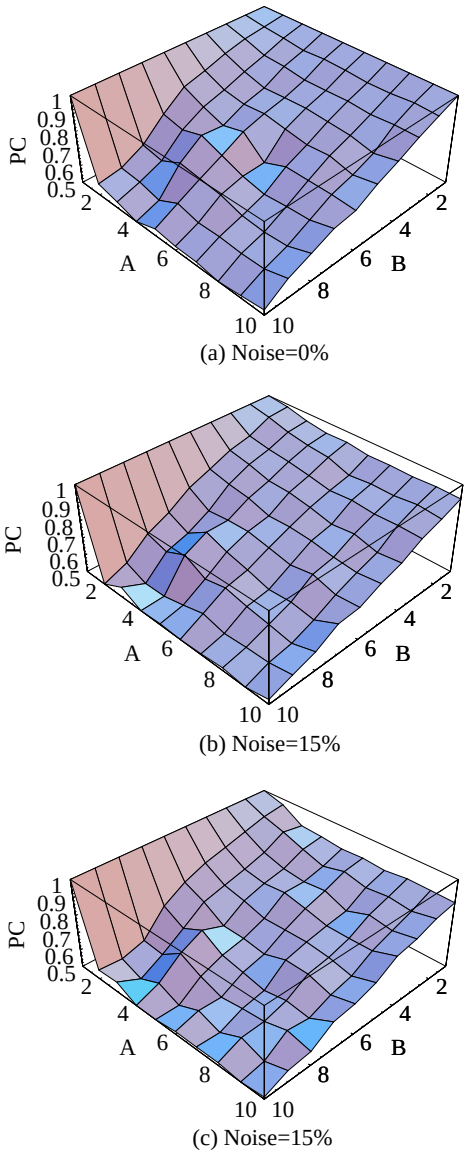

Fig. 1. Results for estimating state sequences from examples of the noisy observation sequences. We have varied each of the A matrix and B matrix from deterministic (axis value=1) to random (axis value = 10). The differences between noise levels are small but most notable along the top right axis.

From Fig. 1(a), it is difficult to determine if the A or B matrix is the most important factor in the HMM. However, if we assume that we need at least an 80% PC rate for the HMM to be useful, we can threshold the results in Fig. 1(a)

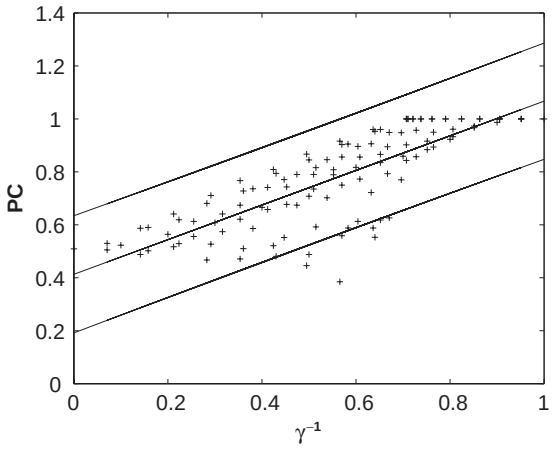

Fig. 2. Shows the relationship between ICN (γ^{-1}) and percentage correct (PC) prediction of observations. The raw data is plotted along with the least squares regression line and the upper and lower 95% confidence interval lines.

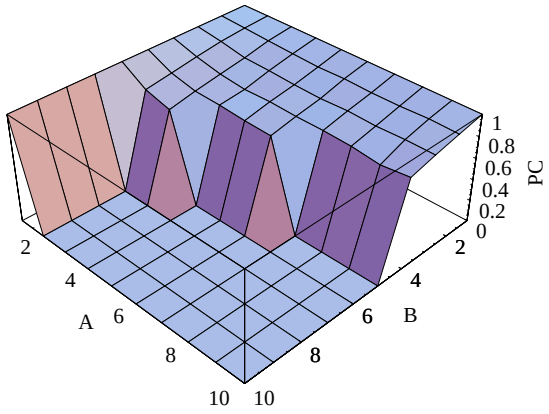

Fig. 3. Results for estimating state sequences with a threshold of 80%. We have varied each of the A matrix and B matrix from deterministic (axis value=1) to random (axis value = 10).

to observe what range of parameters satisfies this criterion. The result is shown in Fig. 3. This figure clearly highlights that the B matrix is the most important factor in the success of an HMM. That is, A is required to be near deterministic to affect the performance over the B matrix—an interesting result since the B matrix can be used to derive the most likely state given the observation and prior state probabilities.

We have performed a similar experiment to those discussed using the $H(v)$ and $H(v \mid b)$ conditional information values as the dependent variables. These results are shown in Fig. 4. As can be seen the entropy of the Viterbi sequence is approximately 1 everywhere which is as expected since we expect the occurrence of both states to be approximately equal in this hypothetical example. Fig. 4(b) clearly indicates that an HMM does no better than a Bayesian classifier when either the observation evidence is very good or the Markovian component approaches the random case. The interesting observation is that this is not a gracefully degrading function. There is a very clear delineation between the areas where the Markovian component is having an effect and those where it has no effect.

Consistent with Fig. 3 the $H(v \mid b)$ results demonstrate the redundancy of the Markov condition when the rank of the B matrix is high: the evidence from observations is unambiguous. In the following, we consider a more realistic application of these measures to assessing just how the components of an HMM-based method for recognizing hand movements contribute to the performance prediction.

4.1.1. Classification performance

What has been discovered about prediction also applies to classification. To illustrate this we have generated a number of HMMs by simultaneously varying both the A and B matrices, as before, from deterministic to uniform(random)as

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0.05 & 0.95 \\ 0.95 & 0.05 \end{bmatrix} \cdots \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}.$$

We selected two HMMs corresponding to: (1) the deterministic case: A and B matrices being the identity with an ICN of 1. (2) the random case, with both A and B matrices having values of 0.5 everywhere and an ICN of 0. For each of these HMMs we then generated a set of observations using HMM Monte Carlo sampling.

The Viterbi algorithm was then run on these observation sequences using the generation (reference) model and the remaining models. MAP values were recorded and plotted against the model average absolute difference to the initial model and the results are shown in Fig. 5. Here model distance was simply defined as the total absolute distance between entries in the model C matrices (Eq. 12). These curves clearly demonstrate that as the HMM contains more uncertainty the identification, via the Viterbi and the MAP criterion algorithm, fails. This also applies to model estimation, using the Baum Welch estimation procedure, though space does not permit a detailed analysis of this.

4.1.2. Iterative results

As a final simulation example, we show how our three measures change with each iteration of the Baum Welch training procedure. In this case we have used a single HMM to generate two observation sequences ($O_{training}$,

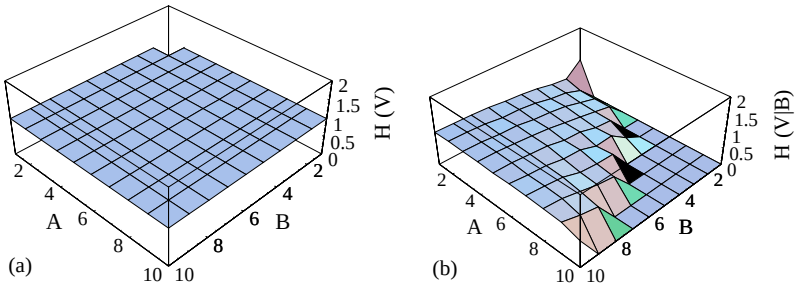

Fig. 4. Shows, as a function the different model parameters. Left: $H(v)$-information in the Viterbi solution. Right: $H(v \mid b)$: the information contained in the the Viterbi solution not contained in the Bayes classifier.

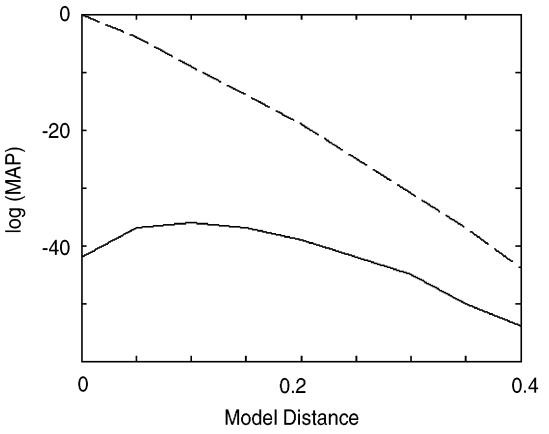

Fig. 5. Shows log(MAP) values as a function of average absolute difference between reference and comparison model parameters. Dashed curve corresponds to the deterministic reference model (Condition Number of 1) and solid curve to the random model (Condition Number of 0).

O_{test}) 10,000 symbols long:

$$A = \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix}, \quad B = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix},$$

$$\pi = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}. \tag{25}$$

From O_{test} we then use Baum Welch to estimate the (known) underlying HMM, initialized by the random HMM:

$$A = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix},$$

$$\pi = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}. \tag{26}$$

Fig. 6(a) shows how the measures vary with each iteration of the Baum Welch estimation procedure, as indexed by log(MAP) values, on the training data. These model estimates at each iteration were then used to compute a Viterbi score (again, log(MAP)) on the test data, at each iteration—as shown in Fig. 6(b) and a comparable percentage correct score (PC) on the test data—Fig. 6(c). We have only shown how the residual sum for state 1 (s_1) changes as training proceeds as we get very similar results for the other symbols (state 2 and observation 1 and 2). The results demonstrate, as expected, how each of the measures improve as training proceeds. In particular, the mutual information measure demonstrates how the Baum Welch estimation procedure quickly settles on a model which exploits both the observation model and the Markov property on the HMM as defined by the model (Eq. (25)).

5. Predicting 3D hand movements from image features

We consider a difficult realistic, yet under constrained, estimation problem—one not uncommon in vision-based gesture prediction. The problem is one of estimating the pitch of a hand from its image. Fig. 7 shows an example sequence of a (synthetic) hand. The motion of the model is rigid about the wrist joint. The poses of the hand are quantized so that there are five possible positions of pitch ($\theta_p = \{-30°, 0°, 20°, 50°, 80°\}$), 5 possible for roll ($\theta_r = \{-90°, -45°, 0°, 45°, 90°\}$) and 4 possible for yaw ($\theta_y = \{-20°, -10°, 0°, 10°\}$) giving a total of 100 possible poses. We have used the aspect ratio of the silhouette as our observation and calculated as the ratio of the smallest to largest eigenvalue of the 2D distribution of the pixels inside the silhouette. Again, the aim was to recover five 3D poses purely from the image under a number of movement conditions corresponding to a deterministic walk, a random walk, and a set of randomly selected poses.

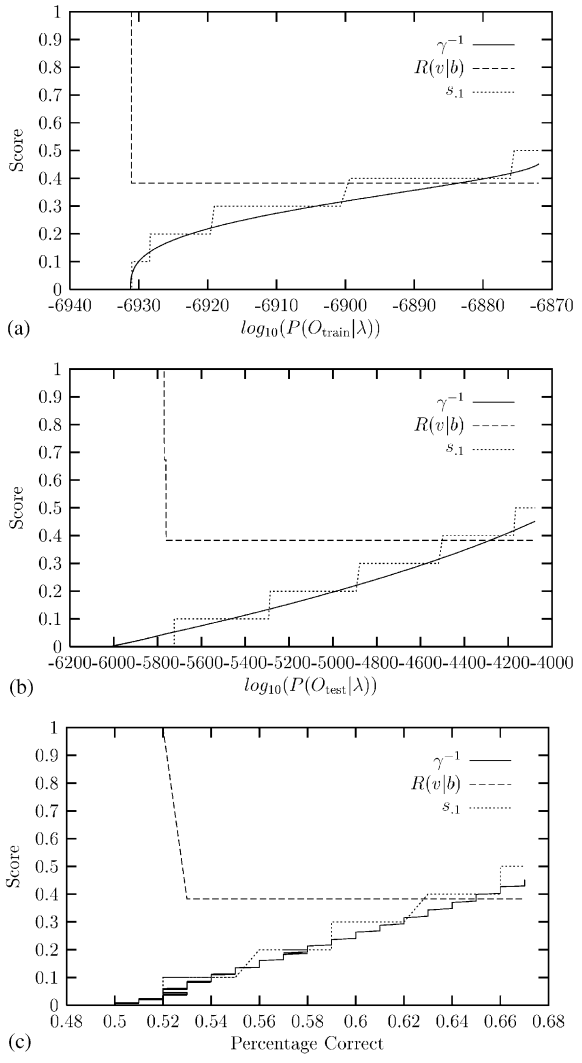

Fig. 6. Results showing how the measures change with each iteration (as indexed by log(MAP) values) of the Baum Welch algorithm on the training data. γ^{-1} is the ICN, $R(v\,|\,b)$ is the residual information of the Viterbi solution given the Bayesian classifier solution, and $s_{.1}$ is the first element of the residual sum vector corresponding to state 1.

5.1. Deterministic walk

In the deterministic walk, the sequence of hand poses was completely predictable. Starting from the neutral pose of the hand, each position of the roll, pitch and yaw is moved to its next position until the maximum range of motion was reached. The motion then reversed in a backward fashion. Fig. 7 shows nine frames of the sequence.

We generated two 1000 length sequences and stored both the ground truth data (i.e. the actual pitch) as well as the

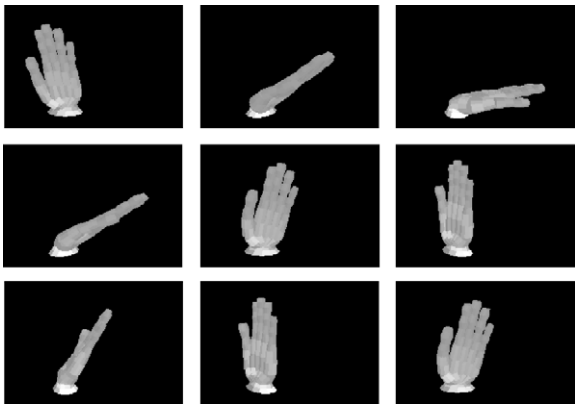

Fig. 7. The gesture training data. Nine sequential (top left to lower right order) frames of a video sequence used in the deterministic movement condition.

observations (the aspect ratio of the silhouette) for each frame of the sequence: one sequence for training and one for testing. Initial estimates of the HMM were then obtained using the moving window method. We then used the Baum Welch procedure. Initially, we partitioned the observation range into five equal aspect ratio ranges. This produced the following HMM:

$$A = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix},$$

$$B = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}. \quad (27)$$

$$\pi = [0.06\ \ 0.13\ \ 0.62\ \ 0.13\ \ 0.06]. \quad (28)$$

If we calculate the ICN of the augmented matrix $A\,|\,B$, we obtain 0.23 indicating that the HMM is not ideal. The residual sum of the matrix is

$$R^1 = [0.5\ \ 0.6\ \ 0.4\ \ 0.6\ \ 0.5\ \ 0.3\ \ 0.9\ \ 0.0\ \ 0.0\ \ 0.0]. \quad (29)$$

This indicates that only the first 2 observation symbols are useful (recall that the first five elements of R^1 refer to the HMM states), and that the rest could be discarded. Clearly R^1_7

(first symbol is R_1^1) performs better than R_6^1 and the overall prediction accuracy is 63%. In turn, the HMM could be improved by refining symbol R_6^1 and removing symbols R_8^1, R_9^1, R_{10}^1. We could possibly also improve the states somewhat by adding new states; however, this is difficult to do meaningfully in a supervised learning situation such as this one. However, in other situations where the states have no particular meaning, state splitting or merging could also be performed as is typically done in previous algorithms for improving HMM topology.

We then split R_6^1 into three distinct symbols (by creating 3 equally sized bins to cover the original) and removed R_8^1, R_9^1 and R_{10}^1 resulting in 4 observation symbols (four new attribute ranges). Re-estimating a new HMM results in

$$B = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}. \quad (30)$$

This gives an ICN of 0.34 and a residual sum vector of

$$R^2 = [0.4\ \ 0.7\ \ 0.5\ \ 0.7\ \ 0.6\ \ 0.4\ \ 0.0\ \ 0.8\ \ 0.9]. \quad (31)$$

This HMM results in a perfect prediction accuracy of 100% on the test sequence. The role of the Markov component of the HMM is measured by the residual information. In this case we found that $H(v) = 2.25$, $H(v \mid b) = 0.75$ and so $R(v \mid b) = 1.5$ or 33% $(100 * H(V/B)/H(V))$ of the information was contained within the A matrix (and the associated Viterbi algorithm) in predicting the optimal state sequence. This result is consistent with the results of the Bayes classifier, alone, which is only 75% correct compared to the complete Viterbi solution of 100% correct.

5.2. Random walk

In the second example, we have performed a random walk over each of the degrees of freedom of the hand: given the current pose there is an equal probability of stepping one step forward or one step backward on each degree of freedom (roll, pitch or yaw). This is a much more difficult problem than the previous one as each pitch pose may occur with any combination of roll or yaw poses. Again estimating an HMM using the moving window technique and then applying the Baum Welch algorithm to produce a final estimate, produces the following HMM with five equally distributed observation symbols:

$$A = \begin{bmatrix} 0.46 & 0.54 & 0.00 & 0.00 & 0.00 \\ 0.45 & 0.00 & 0.55 & 0.00 & 0.00 \\ 0.00 & 0.53 & 0.00 & 0.47 & 0.00 \\ 0.00 & 0.00 & 0.46 & 0.00 & 0.54 \\ 0.00 & 0.00 & 0.00 & 0.55 & 0.45 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.74 & 0.26 & 0.0 & 0.0 & 0.0 \\ 0.90 & 0.10 & 0.0 & 0.0 & 0.0 \\ 0.67 & 0.33 & 0.0 & 0.0 & 0.0 \\ 0.89 & 0.05 & 0.04 & 0.03 & 0.0 \\ 0.86 & 0.14 & 0.0 & 0.0 & 0.0 \end{bmatrix}, \quad (32)$$

$$\pi = [0.20\ \ 0.16\ \ 0.30\ \ 0.14\ \ 0.20]. \quad (33)$$

The ICN for this HMM is 0.13 which indicates correlations and redundancies within the model with a characteristic low prediction performance of 29% correct on the test data. The residual sum is

$$R^1 = [0.4\ \ 0.4\ \ 0.5\ \ 0.4\ \ 0.4\ \ 0.1\ \ 0.1\ \ 0.0\ \ 0.0\ \ 0.0]. \quad (34)$$

Given the relatively better discriminatory power of the observation symbols (last 5 components of R^1) we split the first and second symbols into three new symbols each, leaving the third and fourth symbols and deleting the fifth symbol. The fifth symbol does not appear in the training data and it can be trivially deleted, the third and fourth symbols do appear in the data but very rarely, and we have maintained them for completeness. After rerunning the estimation mode, we obtained an ICN of 0.19, and a residual sum vector of

$$R^2 = [0.5\ \ 0.5\ \ 0.5\ \ 0.5\ \ 0.4\ \ 0.2\ \ 0.2\ \ 0.1\ \ 0.2\ \ 0.1\ \ 0.0\ \ 0.1\ \ 0.1] \quad (35)$$

and prediction performance of 40% correct.

Continuing this process for three more iterations, resulted in an ICN of 0.28 and a prediction performance of 75% correct with 34 observation symbols. The actual matrix is not included for the sake of brevity. In this final case, $H(v) = 2.32$, $H(v \mid b) = 1.59$ and $R(v \mid b) = 0.73$ or 69% of the information was contained in the A matrix in predicting the optimal state sequence. This is confirmed since the ML classifier performed at 50% correct prediction, significantly less than the complete HMM (75%).

5.3. Random poses

For the random poses case, at each frame a random pose for the hand is chosen. Unlike the previous examples, one would expect no contribution from the Markovian element

of the HMM. Using a similar approach to that followed in the above examples (starting with five initial symbols and progressively refining the appropriate ones), after four iterations we arrive at an HMM with an ICN of 0.11 and a residual sum of:

$$R^4 = [0.1\ \ 0.1\ \ 0.1\ \ 0.0\ \ 0.1\ \ 0.3\ \ 0.2\ \ 0.3\ \ 0.1\ \ 0.2\ \ 0.2$$
$$0.1\ \ 0.1\ \ 0.2\ \ 0.1\ \ 0.3$$
$$0.2\ \ 0.1\ \ 0.0\ \ 0.1\ \ 0.1\ \ 0.2$$
$$0.2\ \ 0.1\ \ 0.1\ \ 0.0\ \ 0.0\ \ 0.2\ \ 0.0\ \ 0.1].$$

So clearly, this is an extremely difficult problem. In fact, the prediction performance is 49% correct. In this case we found that $H(v) = 2.06$, $H(v \mid b) = 0.56$ and so $R(v \mid b) = 1.5$ or 27% of the information was contained within the A matrix in predicting the optimal state sequence—indicating that the Markov component of the HMM is not helping and this is confirmed by the performance of the Bayes classifier at 44% correct—quite close to the performance of the full HMM.

6. Recognizing hand gestures

As a further example of the utility of our techniques, we consider the problem of using HMMs to recognize two different classes of hand gestures [11], deictic and symbolic. Deictic gestures are pointing movements toward the left of a body-face space, and symbolic gestures, such as click and rotate, are used to indicate commands. We have used the same data as in [11] which is available at

http://www.idiap.ch/~marcel/Databases/main.html.

The data consists of 2D trajectory information of the relevant gestures normalized to the user's body space. There are many independent observation sequences for each gesture class. Fig. 8 shows a plot of the training data used in this paper. This problem is significantly different than the previous problems we have studied. Specifically, it is a recognition problem rather than an estimation/prediction problem, the input data is two-dimensional and the states are truly hidden. Nevertheless, we can perform a similar analysis as previously, except this time we need to produce 2 HMMs —one for each gesture class. We train each HMM on its relevant training data and then test it using an independent evaluation set as did [11]. At recognition time, an unknown observation sequence is assigned to the most likely class according to the probability of the sequence given each of the two HMMs.

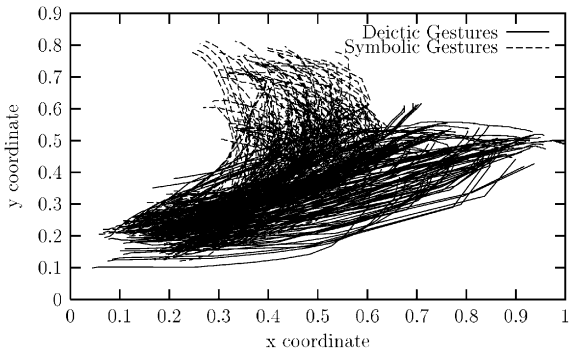

Fig. 8. Distribution of gesture trajectories.

Starting with a simple 2×2 binning of the observation data and 10 hidden states, we choose appropriate observation symbols to split or merge (in a quad-tree fashion) using our residual vector as a guide. In this manner, we proceed from the initial observation symbol map shown in Fig. 9(a) to the map shown in Fig. 9(b). The recognition rate improves from 86.8% to 94.8%, and the ICN for the deictic HMM improves from 0.01 to 0.05, and for the symbolic HMM from 0.02 to 0.06. If we increase the number of hidden states to 20, we can further improve the performance to 97.1%. Curiously, the ICN with 20 states (0.03 and 0.05 for deictic and symbolic respectively) was worse than with 10, although the residual vector indicated that having 20 states was just as appropriate as having 10. The state residual vectors were:

$$R^{10} = [0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.3\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.3]^{\mathrm{T}} \quad (36)$$

and

$$R^{20} = [0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.1\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2$$
$$0.2\ \ 0.2\ \ 0.3\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.2\ \ 0.3]^{\mathrm{T}}. \quad (37)$$

The problem with using the ICN in this case is that it does not take into account the differences between HMMs, but rather treats each HMM independently. Nevertheless, both HMMs are poorly conditioned which questions the two gesture classes defined by Marcel et al. [11]. Even so, our results compare favourably with [11], who reports recognition rates of 98.3% using an Input/Output HMM (IOHMM), rather than our simpler ergodic model.

In terms of information content, for the deictic HMM, we have $H(v)=3.69, H(v \mid b)=2.05$ and so $R(v \mid b)=1.64$ or 56% of the information was contained within the A matrix. For the symbolic HMM $H(v)=2.9$, $H(v \mid b)=1.84$ and so $R(v \mid b)=1.06$ or 63% of the information was contained within the A matrix. Indicating that both HMMs utilize both the Markov chain and the observation data in significant ways.

7. Speech recognition

As a final example, we show how our techniques can be used to aide the choice of HMMs for more complex recognition tasks. In this case we look at a simplified speech recognition task using a subset of the ISOLET database which is available on the web at:

http://www.cslu.ogi.edu/corpora/download/ISOLET_sample.zip

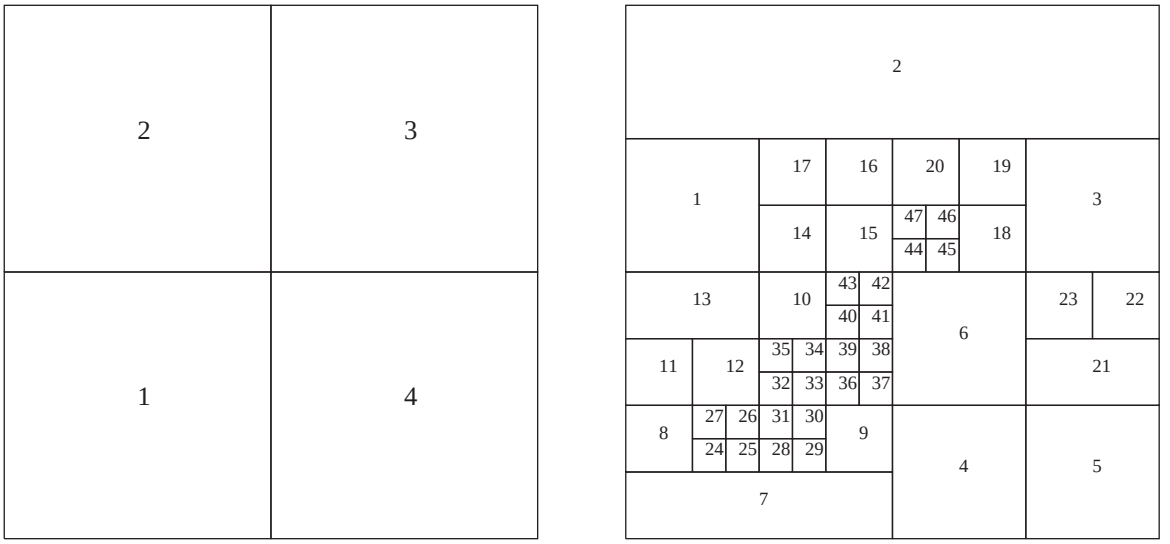

Fig. 9. Left: Initial Observation Map. Right: Final Observation Map derived from the refinement method.

This database contains recordings of 13 speakers uttering each letter of the alphabet twice. The raw data was processed in the same manner as [12]. That is, the speech data was partitioned into 20 ms frames where subsequent frames overlapped by 10 ms. A 28-dimensional feature vector consisting of 14 Mel-scaled FFT cepstral coefficients (MFCC) and their first-order time derivatives were extracted from each frame and these were subsequently clustered into 16 distinct observation symbols. The processing was performed using the publicly available HTK software:

http://htk.eng.cam.ac.uk/

Since the database was small the letters were grouped according to similar sounds in a manner similar to [12]. That is, we used four multi-letter sets: $\{b,c,d,g,p,t,v,z\}$, $\{a,k,j\}$, $\{i,r,y\}$ and $\{m,n\}$, and the remainder of the letters formed individual recognition categories resulting in a 13-class recognition problem. A separate left–right HMM was created for each of the 13 classes and we used the Viterbi log(MAP) score to determine class labels on independent test sequences. We used the first utterance of each speaker for training and the second utterance for testing.

Assuming the observation quantization to be reasonable, then we could only modify the number of states to examine performance. In this case we considered $5, 10, 15, 20, 25, 30$ state models. Although recognition rate was the most important model selection variable, the ICN and mutual information measures were used to discriminate between HMMs with similar recognition rates.

More formally, we have 13 classes, $C_i \in \mathfrak{C}$, each of which has 26 training examples which were split into 13 training ($\mathfrak{S}_{train}(C_i)$) and 13 testing ($\mathfrak{S}_{test}(C_i)$) examples. For the 13 training examples, we trained 6 HMMs, $\lambda^n_{C_i}$, where $C_i \in \mathfrak{C}$ and $n \in \{5, 10, 15, 20, 25, 30\}$ indicates the number of states in the HMM. The recognition rate, $R^n_{C_i}$, of $\lambda^n_{C_i}$ was calculated with respect to other HMMs with the same number of states. That is:

$$R^n_{C_i} = \frac{1}{N} \sum_{e_j \in \mathfrak{S}_{test}(C_i)} \delta_i(F_n(e_j)), \tag{38}$$

where $F_n(e_j)$ is the classification function given by

$$F_n(e_j) = \arg\max_{C_i \in \mathfrak{C}} P(O \mid \lambda^n_{C_i}), \tag{39}$$

and $\delta_i(j)$ is the Kronecker delta function:

$$\delta_i(j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{40}$$

Table 1
Decision variables for choosing the best HMM for character set $\{a,k,j\}$

States	Recognition rate	ICN	$H(v \mid b)/H(v)$
5	0.72	0.300	0.35
10	0.72	0.107	0.38
15	0.90	0.034	0.33
20	0.87	0.037	0.32
25	0.87	0.059	0.34
30	0.87	0.026	0.37

As an example, consider Table 1. First, models with 5 or 10 states were removed due to low recognition rates. The ICN for 25 states was quite significant since we expect the ICN

Table 2
Decision variables for choosing the best HMM for character $\{l\}$

States	Recognition rate	ICN	$H(v \mid b)/H(v)$
5	0.85	0.337	0.26
10	0.77	0.120	0.31
15	0.92	0.027	0.33
20	0.69	0.040	0.31
25	0.77	0.043	0.38
30	0.77	0.035	0.40

to decrease as the number of states increases. For this reason, we might choose the 25 state HMM as the best since the difference over the other variables is insignificant. Table 2 shows another example. The main contenders were the 5 or 15 state HMMs. The mutual information scores do not significantly discriminate between the two, but the ICN is very poor for a 15 state HMM, and hence we would choose the 5 state model as, by definition,the larger the ICN score the more independent are the states and observations to each other. We can continue in this manner to judiciously choose HMMs for each of the character sets. After choosing the "optimal" HMMs, we can calculate a new recognition rate in a manner similar to Eqn. 38 except that the λ_{C_i}'s may have any number of states. This results in an average recognition performance for all classes, C_i, of 0.831 and shows a marginal improvement over simply using the recognition performance as the sole decision variable which results in an average recognition rate of 0.825. The difference is probably not significant for this example, but illustrates that the ICN and mutual information measures can provide a principled method of tie-breaking when other decision variables do not produce significant differences.

8. Discussion

In this paper we have explored three tools for the diagnosis of HMMs. The condition number identifies from the A and B matrices how successful a given HMM is likely to be at generating correct state sequences. The residual sum matrix identifies which states or observation symbols need to be refined or removed from the model to improve it. The last measure based on mutual information identifies if the HMM is likely to do any better than a simple Bayesian classifier (ML) using the B matrix alone. The first and last measures are somewhat independent and identify different (but overlapping) areas of HMM usefulness. The tools are also powerful—they not only identify if a given HMM is useful or not, but also identify exactly what the problem is if an HMM is not performing well.

Furthermore, as demonstrated in Section 5, the practitioner can follow a methodical routine to improve an HMM. It seems quite likely that this pruning and splitting routine could be automated and we intend to pursue this in future. Other avenues to pursue is to extend the analysis to be able to cope with continuous observation densities explicitly (rather than by quantization as in the examples above), and to extend the analysis to coupled HMMs.

We have noted the importance of the matrix $A^{\mathrm{T}}B$ in the analysis of HMMs but did not develop the idea further, focusing instead on the augmented matrix $A \mid B$. We are continuing to investigate the theoretical importance of $A^{\mathrm{T}}B$ and how it can be used to more precisely measure the interaction between the two component matrices.

What is concluded about the prediction or estimation of sequences also holds for the uses of HMMs for temporal pattern recognition and the current measures inform the user as to the degree to which the HMM recognition performance is predictable from the ML classifier, the Markov component, or both.

9. Summary

This paper addresses three problems that are rarely discussed in the use of hidden Markov models for pattern recognition and prediction. One addresses the question as to why a well-fitting HMM does not perform adequately as a classifier or predictor? The second issue, and related to the above question, is the reporting and analysis of the model parameters that fall into two basic components on the HMM: the memory (Markov) model and the observation model. Without knowing these values we show how it can be misleading to conclude anything about whether the HMM is the appropriate way of describing the data. Finally, using standards techniques from Linear Algebra and Information Theory we propose some measures for how these parameters contribute to performance and an algorithm for improving the model topology.

References

[1] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–285.
[2] H. Bunke, T. Caelli, Special edition on HMMs and vision, Int. J. Patt. Recogn. Artif. Intell. 15 (1).
[3] K. Lee, Automatic Speech Recognition: The Development of the SPHINX System, Kluwer Publishers, Dordrecht, 1989.
[4] M. Brand, Structure learning in conditional probability models via an entropic prior and parameter extinction, Neural Comput. J. 11 (5) (1999) 1155–1182.
[5] R. Lyngso, C. Pedersen, H. Nielsen, Metrics and similarity measures for hidden Markov models, in: International Conference on Intelligent Systems for Molecular Biology, 1999, pp. 178–186.
[6] C. Bahlmann, H. Burkhardt, A. Ludwigs, Measuring hmm similarity with the bayes probability of error and its application, in: Sixth International Conference on Document Analysis and Recognition (ICDAR'01), IEEE Computer Society, New York, 2001, pp. 406–411.
[7] V. Balasubramanian, Equivalence and reduction of hidden Markov models, Tech. Rep. A.I. Technical Report No. 1370, MIT Artificial Intelligence Laboratory, 1993.

[8] J. Norris, Markov Chains, Cambridge University Press, Cambridge, UK, 1997.
[9] G. Golub, C. V. Loan, Matrix Computations, 2nd Edition, The Johns Hopkins University Press, Baltimore, MD, 1989, pp. 79–81 (Chapter 2.7).
[10] R. Ash, Information Theory, Interscience Publishers, New York, 1995.
[11] S. Marcel, O. Bernier, J. Viallet, D. Collobert, Hand gesture recognition using input/ouput hidden Markov models, in: FG'2000 Conference on Automatic Face and Gesture Recognition, 2000, pp. 456–501.
[12] A.V. Rao, K. Rose, Deterministically annealed design of hidden Markov model speech recognizers, IEEE Trans. Speech Audio Process. 9 (2) (2001) 111–126.

About the author—DR. BRENDAN MCCANE. He is Senior Lecturer in Computer Science, University of Otago, New Zealand. His current interests lie in Computer Vision, Graphics, Pattern Recognition and Artificial Intelligence and their applications to HCI and Medical Image Understanding. He is a member of the International Association for Pattern Recognition.

About the author—DR. TERRY CAELLI. His interests lie in Computer Vision, Pattern Recognition and Artificial Interlligence and their applications to intelligent sensing, image interpretation and computer assisted perception and action systems. He is Professor of Computing Science at the University of Alberta, Canada. He is a Fellow of the International Association for Pattern Recognition and a Fellow of the Institute for Electronic and Electrical Engineers (IEEE).