

CHAPTER 1

HIDDEN MARKOV MODELS FOR SPATIO-TEMPORAL PATTERN RECOGNITION

Brian C. Lovell^a and Terry Caelli^b

^a*The Intelligent Real-Time Imaging and Sensing (IRIS) Group
The School of Information Technology and Electrical Engineering
The University of Queensland, Australia QLD 4072
E-mail: lovell@itee.uq.edu.au*

^b*National Information and Communications Technology Australia (NICTA)
Research School of Information Sciences and Engineering
Australian National University, Australia
email: tcaelli@ualberta.ca*

The success of many real-world applications demonstrates that hidden Markov models (HMMs) are highly effective in one-dimensional pattern recognition problems such as speech recognition. Research is now focussed on extending HMMs to 2-D and possibly 3-D applications which arise in gesture, face, and handwriting recognition. Although the HMM has become a major workhorse of the pattern recognition community, there are few analytical results which can explain its remarkably good pattern recognition performance. There are also only a few theoretical principles for guiding researchers in selecting topologies or understanding how the model parameters contribute to performance. In this chapter, we deal with these issues and use simulated data to evaluate the performance of a number of alternatives to the traditional Baum-Welch algorithm for learning HMM parameters. We then compare the best of these strategies to Baum-Welch on a real hand gesture recognition system in an attempt to develop insights into these fundamental aspects of learning.

1. Introduction

There is an enormous volume of literature on the application of hidden Markov Models (HMMs) to a broad range of pattern recognition tasks. In the case of speech recognition, the patterns we wish to recognise are spoken words which are audio signals against time. Indeed, the value of Markov models to model speech was recognised by Shannon²⁶ as early as 1948. In the case of hand gesture recognition, the patterns are hand movements in both space and time — we call this a spatio-temporal pattern recognition problem. The suitability and efficacy of HMMs to such problems is undeniable and they are now established as one of the major tools of the pattern recognition community. Yet, when one looks for research which address fundamental problems such as efficient learning strategies for HMMs or perhaps analytically determining the most suitable architectures for a given problem, the number of papers is greatly diminished. So despite the enormous uptake of HMMs since

their introduction in the 1960's, we believe that there is still a great deal of unexplored territory.

Much of the application of HMMs in the literature is based firmly on the methodology popularised by Rabiner *et al.* (1983)^{25,16,24} for speech recognition and these studies are the primary reference for many HMM researchers resulting in two common practices. One, to use the forward algorithm to determine the MAP (maximum posterior probability) of the model, given an observation sequence, as a classification metric. Two, to use the Baum-Welch as a model estimation/update procedure. We will see how these are not ideal strategies to use as, in the former case, classification is reduced to a single number without directly using the model (data summary) parameters, attributes, per se. As for the latter, the Baum-Welch⁴ algorithm (a version of the famous Expectation-Maximisation algorithm^{14,1,21}) is, in the words of Stolke and Omohundro²⁸, "... far from foolproof since it uses what amounts to a hill-climbing procedure that is only guaranteed to find a local likelihood maximum." Moreover, as observed by Rabiner²⁴, results can be very dependent on the initial values chosen for the HMM parameters.

The problem of finding local rather than global maxima is encountered in many other areas of learning theory and optimisation. These problems are familiar territory to researchers in the artificial neural network community and many techniques have been proposed to counter them. Moreover genetic and evolutionary algorithmic techniques specialise in solving such problems — albeit often very slowly, especially in the case of biological evolution¹¹. With this in mind, we use simulated data to investigate other approaches to learning HMMs from observation sequences in an attempt to find superior alternatives to the traditional Baum-Welch Algorithm. Then we compare and test the best of the alternate strategies on real data from a hand gesture recognition system to see if the real data trials corroborate the conclusions drawn from simulated trials.

1.1. Background and Notation

In this study, we focus on the discrete HMM as popularised by Rabiner²⁴. Using the familiar notation from his tutorial paper, a hidden Markov model consists of a set of N nodes, each of which is associated with a set of M possible observations. The parameters of the model include an initial state vector

$$\pi = [p_1, p_2, p_3, \dots, p_N]^T$$

with elements p_n , $n \in [1, N]$ which describes the distribution over the initial node set, a transition matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}$$

with elements a_{ij} with $i, j \in [1, N]$ for the transition probability from node i to node j

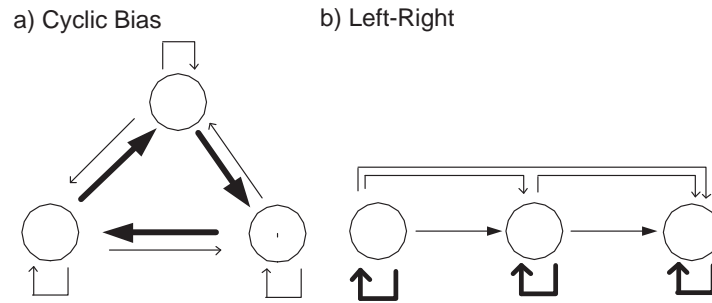


Fig. 1. Cyclic and Left-Right structures. Bold arrows indicate high probability transitions. No arrow between vertices indicates a forbidden (zero-probability) transition.

conditional on node i , and an observation matrix

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1M} \\ b_{21} & b_{22} & \dots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \dots & b_{NM} \end{pmatrix}$$

with elements b_{im} for the probability of observing symbol $m \in [1, M]$ given that the system is in state $i \in [1, N]$. We denote the HMM model parameter set by $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$.

The model order pair (N, M) together with additional restrictions on allowed transitions and emissions defines the topology or structure of the model (see figure 1 for an illustration of two different transition structures). One commonly used topology is called Fully-Connected (FC) or Ergodic. In the FC HMM there is not necessarily a defined starting state and all state transitions are possible such that $a_{ij} \neq 0 \forall i, j \in [1, N]$. Another topology, especially popular in speech recognition applications, is called Left-Right. In an LR HMM there is a defined starting state (usually state 1) and only state transitions to higher-index states are allowed such that $a_{ij} = 0 \forall i > j$ where $i, j \in [1, N]$.

Rabiner²⁴ defines the three basic problems of HMMs by:

Problem 1 Given the observation sequence $O = O_1 O_2 \dots O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence given the model?

Problem 2 Given the observation sequence $O = O_1 O_2 \dots O_T$, and the model λ , how do we choose a corresponding state sequence $Q = q_1 q_2 \dots q_T$ which is optimal in some meaningful sense (*i.e.*, best “explains” the observations)?

Problem 3 How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

Problems 1 and 2 are elegantly and efficiently solved by the forward and Viterbi^{29,12} algorithms respectively as described by Rabiner in his tutorial. The forward algorithm is used to recognise matching HMMs (*i.e.*, highest probability models, MAP) from the observation sequences. Note, again, that this is not a typical approach to pattern classification as

it does not involve matching model with observation attributes. That would involve comparing the model parameters and estimated observation model parameters. MAP does not perform this and so it cannot be as sensitive a measure as exact parameter comparisons. Indeed, a number of reports have already shown quite different HMMs can have identical emissions(observation sequences) ^{18,3}. The Viterbi algorithm is used less frequently as we are normally more interested in finding the matching model than in finding the state sequence. However, this algorithm is critical in evaluating the precision of the HMM; in other words, how well the model can reconstruct (predict) the observations.

Rabiner proposes solving Problem 3 via the Baum-Welch algorithm which is, in essence, a gradient ascent algorithm — a method which is guaranteed to find local maxima only. Solving Problem 3 is effectively the problem of learning to recognise new patterns, so it is really the fundamental problem of HMM learning theory; a significant improvement here could boost the performance of all HMM based pattern recognition systems. Therefore it is somewhat surprising that there appear to be relatively few papers devoted to this topic — the vast majority are devoted to applications of the HMM. In the next section we compare a number of alternatives to and variations of Baum-Welch in an attempt to find superior learning strategies.

2. Comparison of Methods for Robust HMM Parameter Estimation

We focus on the problem of reliably learning HMMs from a small set of short observation sequences. The need to learn rapidly from small sets arises quite often in practice. In our case, we are interested in learning hand gestures which are limited to just 25 observations. The limitation arises because we record each video at 25 frames per second and each of our gestures takes less than one second to complete. Moreover, we wish to obtain good recognition performance from small training sets to ensure that new gestures can be rapidly recognised by the system.

Four HMM parameter estimation methods are evaluated and compared by using a train and test classification methodology. For these binary classification tests we create two random HMMs and then use each of these to generate test and training data sequences. For normalization, we ensure that each test sequence can be correctly recognized by its true model; thus the true models obtain 100% classification accuracy on the test data by construction. The various learning methods are then used to estimate the two HMMs from their respective training sets and then the recognition performance of the pair of estimated HMMs is evaluated on the unseen test data sets. This random model generation and evaluation process is repeated 16 times for each data sample to provide meaningful statistical results.

Before parameter re-estimation, we initialize with two random HMMs which should yield 50% recognition performance on average. So an average recognition performance above 50% after re-estimation shows that some degree of learning must have taken place. Clearly if the learning strategy can perfectly determine both of the HMMs which generated the training data sets, we would have 100% recognition performance on the test sets.

We compare four learning methods 1) traditional Baum-Welch, 2) ensemble averaging

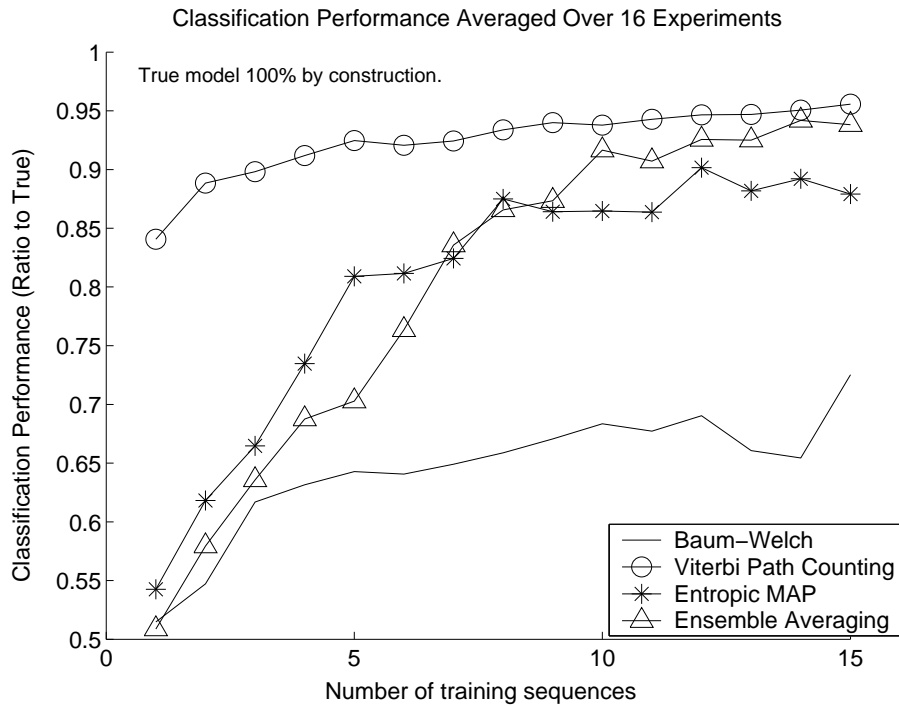


Fig. 2. Relative performance of the HMM parameter estimation methods as a function of the number of training sequences. Viterbi Path Counting produces the best quality models with a much smaller number of training iterations.

introduced by Davis and Lovell⁹ based on ideas presented by Mackay¹⁹, 3) Entropic MAP introduced by Brand⁶, and 4) Viterbi Path Counting¹⁰ which is a special case of Stolke and Omhundo's Best-First algorithm²⁸. The results in figure 2 indicate that these alternate HMM learning methods all classify significantly better than the well-known Baum-Welch algorithm and also require less training data. The Entropic MAP estimator performs well but surprisingly the performance is much the same as simple ensemble averaging. Ensemble averaging involves training multiple models using the Baum-Welch algorithm and then simply averaging the model parameters without regard to structure. Note that for a single sequence, ensemble averaging is identical to the traditional usage of the Baum-Welch algorithm. Overall, the stand-out performer was the VPC algorithm. In these and other trials, this method converges to good models very rapidly and has performed better than the other methods in virtually all of our simulated HMM studies.

3. Video Gesture Recognition

In an attempt to corroborate the strong performance of VPC compared to Baum-Welch on a real-world application, we test various learning techniques on a system for real-time video gesture recognition as shown in figure 3.

In earlier related work, Starnier and Pentland²⁷ developed a HMM-based system to recognise gesture phrases in American Sign Language. Later, Lee and Kim¹⁵ used HMM-based hand gesture recognition to control viewgraph presentation in data projected seminars. Our system recognizes gestures based on the letters of the alphabet traced in space in front of a video camera. The motivation for this application is to produce a way of typing messages into a camera-equipped mobile phone or PDA using video gestures instead of the keypad or pen interface. We use single stroke letter gestures similar to those already widely used for pen data entry in PDAs. For example, figure 3 shows the hand gestures for the letters “Z” and “W.” The complete gesture set is shown in figure 6.



Fig. 3. “Fingerwriting:” Single stroke video gesture for letters “W” and “Z.”

Each video sequence comprises 25 frames corresponding to one second of video. Skin colour segmentation in YUV colour space is applied to locate the hand. Pre-processing (morphological) operations smooth the image and remove noise before tracking the hand with a modified Camshift algorithm⁵. After segmenting the hand, we calculate image moments to find the centroid in each frame. Along the trajectory, the direction (angle) of motion of each of the 25 hand movements is calculated and quantized to one of 18 discrete symbols. The resultant discrete angular observation sequence is input to the HMM classification module for training and recognition.

We compare traditional Baum-Welch with the most promising alternative from the stimulated study, VPC. We evaluate recognition performance over all 26 character gestures using fully connected (FC), left-right (LR), and left-right banded (LRB) model topologies with the number of states ranging from 1 to 14. A LRB model is an LR model which has a transition structure containing self-transitions and next state transitions only (*i.e.*, states cannot be skipped) as shown in figure 5. More formally, $a_{ij} \neq 0 \forall j = i \text{ or } j = i + 1$, and 0 otherwise, $i, j \in [1, N]$.

Our video gesture database contains 780 video gestures with 30 examples of each gesture. Recognition accuracy is evaluated using threefold cross-validation where 20 gestures are used for training and 10 for testing in each partition. These HMMs are initialized with random HMM parameters before using either Baum-Welch or VPC for learning.

From figure 4 the best average recognition accuracy achieved is 97.31% when VPC is used for training, topology is LRB, and the number of states is 13. Although this corroboration

Number of States	Baum-Welch			VPC		
	FC	LR	LRB	FC	LR	LRB
1	80.00	80.00	80.00	80.38	80.38	80.38
2	72.69	94.23	93.85	71.15	91.92	90.77
3	66.54	92.31	96.15	63.85	91.15	93.08
4	80.00	84.80	85.38	53.20	91.20	90.38
5	75.20	81.20	90.77	59.60	91.20	95.00
6	75.60	84.80	85.77	55.20	90.40	93.85
7	77.60	86.40	89.62	45.60	91.20	94.23
8	76.80	86.00	89.62	44.40	90.40	94.23
9	77.60	85.60	90.00	49.20	90.40	94.62
10	76.00	81.60	88.46	43.20	90.00	95.00
11	65.20	86.80	89.23	42.80	90.00	95.00
12	74.80	86.80	88.08	40.80	90.00	95.77
13	84.80	84.00	90.00	39.60	90.00	97.31
14	72.80	81.60	88.46	38.80	90.40	93.46
Mean	75.40	85.44	88.96	51.98	89.90	93.08
Max	84.80	92.31	96.15	63.85	91.20	97.31

Fig. 4. Average percent correct recognition for all 26 video letter gestures against topology and training method.

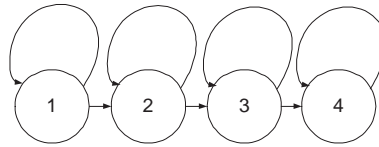


Fig. 5. Left-Right banded topology.

rates the stronger VPC performance exhibited in our simulated data performance trials, a closer investigation of Table 4 raises some doubts about this conjecture through the following observations.

- The Baum Welch algorithm did almost as well as VPC with a best performance of 96.15% correct recognition with only 3 states. Moreover we achieve a very surprising 80% correct recognition with just a single state.
- Topology (*i.e.*, constraints on the initial value of the A matrix) has more impact on performance than the choice of learning algorithm.
- Good recognition performance can be obtained over a very broad range of N , the number of states.

3.1. Comments on Learning Algorithm Performance

We do not suggest that the above observations can be generalized to other real-world application domains but anecdotal evidence from other researchers suggests that similar be-

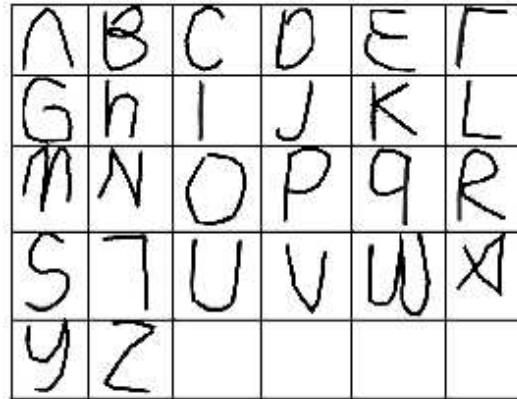


Fig. 6. The alphabet of single-stroke letter hand gestures.

haviour is often encountered. When we designed this gesture system, we thought that this pattern recognition problem was quite challenging and would significantly differentiate learning strategies. Yet the surprisingly good performance over a number of learning algorithms, topologies, and a broad range N suggests that the problem is significantly easier than we suspected.

Our intuition suggests that 3 states is far too small a number to adequately model all of these complex letter gestures, but results show that it is indeed possible to find a three state HMM which yields very good recognition performance. We conjecture that the observation matrix B seems to provide most of the recognition performance and that recognition may be only weakly affected by good estimation of the transition matrix A .

In support of this idea, we may consider the following interpretation of the HMM. Consider each row of the B matrix as the probability mass function of the observation symbols emitted in a given state. In the limiting case of a single state HMM, the B matrix becomes a vector of source symbol probabilities and application of the forward algorithm for recognition is thus equivalent to the well-known and powerful MAP classifier. Indeed from figure 4, we see that this single state degenerate HMM can achieve 80% recognition performance. So sometimes even if the state transitions are poorly modelled, it is quite possible to find good classifiers based on source statistics.

Now clearly if three states can yield strong performance, good HMMs with more than three states must also exist — a simple way to prove this is to note that we can always add additional states which are unreachable (*i.e.*, transition probability of zero) without affecting recognition performance. This may help explain why performance stays much the same over a broad range of N as we increase N beyond three.

The question that arises is, “Why does the Baum-Welch algorithm perform so well on real-world data despite its theoretical flaws and rather poor performance on the simulated HMM data?” Once again, a possible explanation is that this particular spatio-temporal recognition task is relatively easy, so all methods can do quite well. This conjecture may be

true for other common HMM applications and is a focus of current research. Unfortunately, unlike simulated data, the effort of gathering very large and diverse databases of real-world pattern recognition problems to evaluate the performance of different training algorithms is immense, so progress is slow.

3.2. Comments on Topology

The FC topology allows transitions from any state to any other state and does not have a defined starting state. Being the most general topology, it is hardly surprising that performance is relatively poor. In this case we are required to search for a good solution in a parameter space of much higher dimensionality than for LR — so it would be much harder to locate globally optimal solutions. Gestures have a natural start and finish and thus it is reasonable to adopt the LR model as used in speech recognition to great effect.

An even simpler topology is the LRB HMM where only self-transitions and next state transitions are allowed. Thus the A matrix is of the form:

$$\mathbf{A} = \begin{pmatrix} a_{11} & 1 - a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & 1 - a_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{NN} \end{pmatrix} \quad (1)$$

In this case the expected number of observations, \bar{n} , (*i.e.*, duration) in state i is simply given by²⁴

$$\bar{n} = \frac{1}{1 - a_{ii}}. \quad (2)$$

One can interpret the LRB HMM as being an adjustable clock that ideally synchronizes the changes in observation statistics with the changes in state to produce a time-variant MAP classifier. In many ways this HMM topology may be considered as a form of dynamic time-warping²² — a earlier technique used in speech recognition that has fallen out of favour since the advent of HMMs. The good performance of LR and LRB topologies on the gesture data set help make the point that simple HMM topologies often work best on real data.

4. Direct Calculation of HMM Parameters from Video Gestures

A major dissatisfaction with the foregoing treatment of HMM learning strategies on real data is that the learning procedure is be treated like a black-box with little real insight into the learning process. In contrast to simulated data, we are unlikely to know the true HMM that originally generated the data — indeed real data is very rarely generated by a process that even remotely resembles a HMM. Thus we usually don't really know the best topology or number of states *a priori* — practitioners just try ranges of values and pick the one that yields good classification performance. Sometimes, as in the above example, this utilitarian method can find HMMs that are far too simple to accurately characterize the

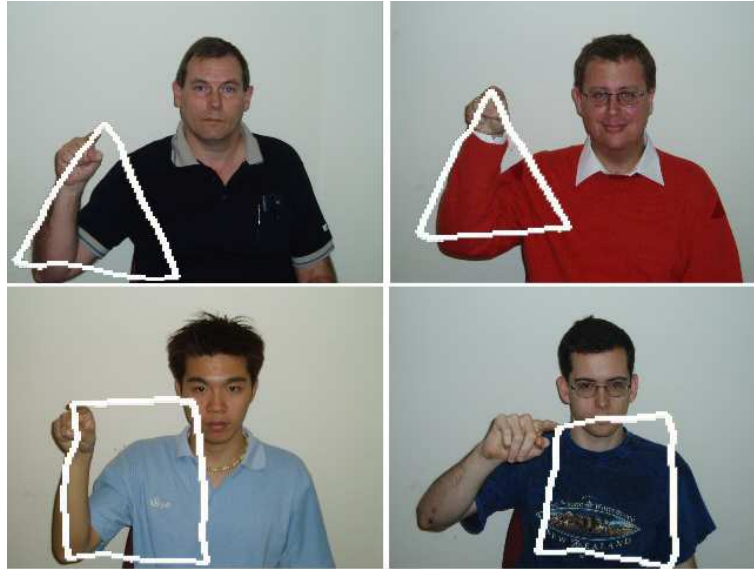


Fig. 7. Triangle and square gestures for direct computation.

state transitions of the underlying pattern. Alternatively, we may find HMMs that are far more complicated than is necessary. In either case this may lead to poorer generalization ability than might otherwise have been the case.

To investigate this topic further, we devised two simple video gestures as shown in figure 7 where it is possible to approximately determine the form of the true HMM directly from the gestures themselves. In the case of the triangle gesture, this can be modelled by a 3-state HMM — each state corresponding to one side of the triangle. There is no need for skipped states to model this gesture, so the LRB topology is the most appropriate.

4.1. Analytic Calculation of A and B

The triangle gesture is intended to be an equilateral triangle, so the expected number of observations in each state is equal and corresponds to $\bar{n} = 24/3 = 8$. Thus the expected values of the transition matrix (A) parameters can be calculated from the duration equation (2).

The triangle gesture consists of a horizontal movement at angle 0 degrees, followed by movement of 120 degrees and then -120 degrees for the other two sides of the triangle. To account for variations in the real gestures, one could use an error distribution (eg., Gaussian) centred on each of these angles as the expected value of each corresponding row of the B matrix.

Similar analysis can be used to derive the expected form of the A and B for the square gesture.

4.2. Re-estimation of A matrix using Baum-Welch after correct initialization

Figure 8 show the calculated values for the A matrix for both gestures as well as the average value over 20 trials after Baum-Welch re-estimation using the calculated value as the initial value for A and a random matrix as the initial value for B . Clearly, the application of Baum-Welch re-estimation did not significantly alter the A matrix which implies that a local maximum was reached near the calculated value of A as expected. Nevertheless, when we used the traditional approach of initial random values for both A and B , we achieved values close to the calculated values only about 50% of the time. Thus even for these simple patterns, it seems that traditional application of the Baum-Welch algorithm with random initial starts has difficulty in finding the transition structure.

A Matrix	Triangle			Square			
Calculated	0.87	0.13	0	0.83	0.17	0	0
	0	0.87	0.13	0	0.83	0.17	0
	0	0	1	0	0	0.83	0.17
				0	0	0	1
Average after BW re-estimation	0.87	0.13	0	0.85	0.15	0	0
	0	0.87	0.13	0	0.83	0.17	0
	0	0	1	0	0	0.85	0.15
				0	0	0	1

Fig. 8. Calculated A matrix and average values after 20 trials of Baum-Welch re-estimation for triangle and square gestures.

Figure 9 shows the performance of a number of alternate strategies to learning HMMs. We measure the sum of the squares of the differences of the elements between the A and B matrices obtained from the Baum-Welch algorithm and our reference matrices which are our best estimate of the correct HMM. In the case of the A matrix we use the calculated value from subsection 4.1 as the reference. For the B matrix, we could determine the reference B matrix in a number of ways (eg., histogram of observation symbols in each state, fitted Gaussian distribution, fitted Von Mises Distribution). Here we choose to use the histogram method.

In methods 1 and 2, we keep A fixed at the calculated reference value and allow B to be re-estimated by Baum-Welch. In method 1, B is initialized to a random value, but in method 2 it is initialized to the reference value. Method 3 is traditional Baum-Welch where both A and B are re-estimated from random initial values. Finally, in method 4 we use Baum-Welch with both matrices initialized to the reference values.

From this table it is quite clear that Baum-Welch is not very good at finding the true HMM parameters even if it is given the topology (A) through a calculated A matrix as in method 1. However when provided with good estimates for both A and B as initial values as in method 4, Baum-Welch converges to the given solution indicating that a local maximum has indeed been reached.

Method: Initialization	A triangle	B triangle	A square	B square
1. A fixed: A calc, B random	0	0.4654	0	0.7284
2. A fixed: A calc, B calc	0	0.0066	0	0.0698
3. A re-est: A random B random	0.0708	0.8632	0.1314	1.437
4. A re-est: A calc B calc	2.13e-5	0.0059	8.19e-4	0.0676

Fig. 9. Average squared error of matrices over 20 trials using different Baum-Welch re-estimation techniques and initial values.

4.3. Comments on HMM learning

The foregoing indicates that the Baum-Welch algorithm may have great difficulty in finding solutions that closely match the transition structure of the physical systems even for very simple examples. Nevertheless, it is clear that the resulting HMMs may still be extremely useful for pattern recognition. We conjecture that this may be because many real-world problems are adequately distinguished by their source statistics so the recognition problem is simpler than it would first appear.

5. Tools for Investigating HMM Parameters and Performance

Such observed anomalies in HMM learning behaviour led us propose a methodology and tools to diagnose and analyze HMM pattern recognition systems. Too often researchers simply report “a HMM with x -states and y -symbols successfully classified the observation sequences”. What does this mean? Is this really the case? Could it have been simply due to excellent discriminating observation attributes (the B -matrix) or a very strong memory model (A -matrix)? Since these matrices are typically not reported very little can be concluded on such issues. Further, a set of HMM models may well be sufficient for classifying a subset of observation sequences but if the A and B matrix probabilities demonstrate high uncertainty the classification performance can easily degrade as many different sets of observations can produce the same, often low, MAP scores. In order to be able to reproduce past results, examine model generalisations, and update model topology the model parametric values require interpretation.

We^{7,20} have made some initial studies into such issues and, in particular, have developed some measures of performance of a given HMM that allows for model refinement and some objective measures of how the observation (B) and state (A) models contribute to performance. Consider the row-augmented matrix:

$$C = A|B. \quad (3)$$

Each row represents all the information about a given state at time t in terms of expected state transitions and observation likelihoods. Consequently the rank of this matrix is critical in understanding the uniqueness or redundancy of states and 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 eigenvalues²³. When all rows of the Markov Chain are identical the Markov Chain condition breaks down in so far as:

$$P[\vec{S}_{t+1}|\vec{S}_t] = P[\vec{S}_{t+1}] = P[\vec{S}_t] = \pi. \quad (4)$$

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:

$$P[O_k|S_i] = P[O_k]. \quad (5)$$

In all then, using the singular values of C , σ , the Inverse Condition Number (ICN) of C ¹³ is:

$$\gamma^{-1} = \sigma_{min}/\sigma_{max} \quad (6)$$

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.

By projecting individual rows of C into it’s eigenspace we can then observe the redundancy of given states (also evident in the row correlation matrix) and consequently split and merge states to update the model topology to minimize model redundancy with respect to both state and observation parameters. This we found to be successful in the case of gesture and speech recognition experiments (see [20] for details).

Given that these above techniques can be used to analyse and refine HMM model topologies — independent of any observation sequence — we now consider how the A and B parameters contribute to the prediction of state sequences given a model *and* observations and we show how Conditional Information² can be used for this purpose. If the B matrix is unambiguous (for example, orthogonal with an ICN of 1.0) then a direct use of either Maximum Likelihood (ML: $\max_S \{P[O(t)|S]\}$) or maximum posterior probability (MAP: $\max_S \{P[S|O(t)] = P[O(t)|S]P[S]\}$) 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.

Given a model and an input observation sequence two state sequences can be generated, 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:

$$H(v|b) = H(v, b) - H(b) \quad (7)$$

where

$$H(v, b) = - \sum_{i,j} (P(S_v = i, S_b = j) \log P(S_v = i, S_b = j)) \quad (8)$$

and

$$H(b) = - \sum_j (P(S_b = j) \log P(S_b = j)). \quad (9)$$

$H(v|b)$ 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

$$R(v|b) = H(v) - H(v|b) \quad (10)$$

provides a measure of how much information the A matrix, and the associated Viterbi algorithm, add to the complete optimal state sequence prediction. These measures provide a clear measure of the degree of information contributed by the observation and state models in the (MAP-based) performance of any HMM on a given data set.

6. Conclusions

HMMs are an immensely powerful tool for solving pattern recognition and classification problems. Many studies demonstrate that it is a powerful technique, but few studies give any insight into why the performance is so good. It is well-known that the Baum-Welch algorithm is a hill-climbing technique that is generally unable to find global maxima. Yet the performance of pattern recognition systems based on Baum-Welch is often very good. Although we can find HMM training algorithms that appear to perform much better than Baum-Welch on simulated HMM classification data, these results do not necessarily translate into greatly improved performance in real-world applications. We conjecture that that is possibly because many important real-world problems have little need for highly complex HMMs. Also it appears that embedding knowledge of the topology of the system can often improve recognition performance more significantly than changes in learning algorithm. Finally we offer some tools and methodologies to analytically investigate and diagnose these problems.

7. Acknowledgements

This paper covers a number of research themes currently being studied within our research groups and collaborators. I would like to acknowledge the contributions of Brendan McCabe, Peter Kootsookos, Richard Davis, Nianjun Liu, and Christian Walder.

References

1. A. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
2. R. Ash. *Information Theory*. Interscience Publishers, 1995.
3. V. Balasubramanian. Equivalence and reduction of hidden markov models. Technical Report A.I. Technical Report No. 1370, MIT Artificial Intelligence Laboratory, 1993.
4. L. Baum, T. Petrie, T. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41:164–171, 1970.
5. G. R. Bradski. Computer video face tracking for use in a perceptual user interface. *Intel Technology Journal*, Q2, 1998. last visited May 2004.
6. M. Brand. An entropic estimator for structure discovery. *Advances in Neural Info. Proc. Systems*, 11:723–729, 1999.
7. T. Caelli and B. McCabe. Components analysis of hidden markov models in computer vision. In *Proc. of 12th International Conference on Image Analysis and Processing*, pages 510–515.
8. Terry Caelli, Andrew McCabe, and Garry Briscoe. Shape tracking and production using hidden markov models. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 15(1):197–221, February 2001.
9. R. I. A. Davis, B. C. Lovell, and T. Caelli. Improved estimation of hidden markov model parameters from multiple observation sequences. In *Proceedings of the International Conference on Pattern Recognition (ICPR2002)*, volume 2, pages 168–171, Quebec City, August 2002. IEEE.
10. Richard I. A. Davis and Brian C. Lovell. Comparing and evaluating hmm ensemble training algorithms using train and test and condition number criteria. *Pattern Analysis and Applications*, 6(4):327–336, February 2003.
11. Richard Dawkins. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*. W. W. Norton, New York, 1996.
12. G. D. Forney. The viterbi algorithm. *Proc. IEEE*, 61:268–278, March 1973.
13. G. Golub and C. Van Loan. *Matrix Computations*, chapter 2.7, pages 79–81. The Johns Hopkins University Press, 2nd edition, 1989.
14. H. Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194, 1958.
15. H.-K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 21(10):961–973, October 1999.
16. S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *The Bell System Technical Journal*, 62(4):1035–1074, 1983.
17. Nianjun Liu, R. I. A. Davis, B. C. Lovell, and P. J. Kootsookos. Effect of initial hmm choices in multiple sequence training for gesture recognition. In *Proceedings of the International Conference on Information Technology*, volume 1, pages 608 – 613, Las Vegas, Nevada USA, April 5-7 2004.
18. R.B. Lyngso, C.N. Pedersen, and H. Nielsen. Metrics and similarity measures for hidden markov models. In *International Conference on Intelligent Systems for Molecular Biology*, pages 178–186, 1999.

19. D. J. C. Mackay. Ensemble learning for hidden markov models. Technical report, University of Cambridge, 1997.
20. B. McCane and T. Caelli. Diagnostic tools for evaluating hmm components. Technical report, Department of Computer Science, University of Otago, 2001.
21. G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley series in probability and statistics. John Wiley and Sons, 1997.
22. C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, September 1981.
23. J. Norris. *Markov Chains*. Cambridge University Press, Cambridge, England, 1997.
24. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
25. L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey, 1993.
26. C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, July 1948.
27. T. Starner and A. Pentland. Real-time american sign language recognition. *IEEE Trans, On Pattern Analysis and Machine Intelligence*, 20:1371–1375, December 1998.
28. A. Stolke and S. M. Omohundro. Best-first model merging for hidden markov model induction. Technical report, International Computer Science Institute, April 1994. TR-94-003.
29. A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, April 1967.