Components analysis of hidden Markov models in Computer Vision

Terry Caeli
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
tcaelli@ualberta.ca

Brendan McCane
University of Otago
Dunedin
New Zealand
SecondAuthor@institution2.com

Abstract

Hidden Markov models (HMMs) have become a standard tool for pattern recognition in computer vision. However, issues of parameter estimation and evaluation are rarely addressed though they play key roles in just how HMMs perform. Without addressing these issues it can be readily shown that a so-called HMM model may actually be a Bayesian classifer or Markov Chain. In this paper we develop methods for addressing issues of assessing HMM component and parameter contributions and illustrate these issues in a representative task of gesture recognition - 3D motion recovery from 2D projections.

1 Introduction

Over the past 20 years hidden Markov models (HMMs) have provided a structural pattern recognition model where the aim has been to infer sequences of underlying or "hidden" states from time-varying signals. The states are assumed to obey a first-order Markov condition while the signal is typically encoded by a discrete set of observation "symbols". The Viterbi algorithm has been the most popular method for predicting optimal hidden state sequences and it's associated maximum posterior probability (MAP) score is typically used for temporal pattern recognition, classification. HMMs have seen particular interest in character, handwriting and gesture recognition as well as many other spatio-temporal visual pattern recognition domains. In these cases different HMM topologies have been developed from simply to factorial, hierarchical and coupled HMMs.

Following the HMM nomenclature of Rabiner [6] a discrete HMM, 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:

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

$B = \{b_j(k)\}$ is an $N \times M$ matrix which is the probability distribution of observation symbol, o, given state j:

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

and $\pi = \{\pi_i\}$ is the initial state distribution where

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

Factorial, hierarchical and coupled HMMs typically involve augmented A matrices to accommodate for additional intra and inter state dependencies as well as additional state-dependent observation, and even observation-to-observation, models[6, 3, 2].

In practice the following procedures and measures have typically been used.

- For estimation, Expectation-Maximization (EM, the Baum Welch algorithm) still remains the most common parameter estimation model. Here, an initial model (typically randomly chosen) is updated or conditioned by given observation sequences for each model. EM is well known to fall into local minima and it is by no means clear what is the best estimation strategy, even using EM, when there are multiple observation sequences.

- For pattern recognition the MAP Viterbi score is the most common decision criterion for selecting between candidate models[6]. We will see how this criterion is less than optimal for models with high uncertainty.

- For prediction, Viterbi-type (dynamic programming) look-ahead is typical.

In addition to the above noted reservations about these procedures, there are also a number of additional issues that address the actual role of the model components - particularly as a function of their values. Consider the three state transition matrices,A, and three state-dependent observation B matrices (2 state and 2 observation symbol cases)

$$A_1 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.8 & 0.2 \end{bmatrix} \quad A_3 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$$

and

$$B_1 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0.8 & 0.2 \\ 0.8 & 0.2 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}.$$

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 a "HMM" model. This is because in estimation, recognition and prediction the key evidence combination function at every time step is $a_{ij}b_{jk}(O_t)$ for states i, j and observation symbol k.

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[5], 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 with no ability to discriminate between any pair of such sequences.
- One with A_1, B_2 would also fail in prediction of the optimal state sequence due to the correlation between the evidence for states given observations 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 purely based on the current evidence at any time step - based purely on the B matrix acting purely as a *Bayesian (ML or MAP) classifier* - see below and Eqns. 2 and 3 above.
- 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.

These simple examples indicate the importance of the model parameters in determining performance, discriminating between models. This paper focuses on how to objectively determine two key HMM component dimensions: ambiguity and uncertainty - and we define these terms via Condition Number, Residual Sum Vector, and Conditional Information content, as follows.

1.1 Condition Number of a HMM

Experienced users of HMMs know that the best performing HMMs are those for which the rows of the A and B matrices are linearly independent for maximum discrimination of state and observation symbols. However, since A and B are inextricably linked in the model execution, it makes sense to define the following row augmented matrix:

$$C = A|B. \tag{4}$$

where each row provides the complete description of a given state, except for its steady state probability, explicitly. Accordingly, to minimize model state and symbol ambiguity we would like the rows of C to be as linearly independent as possible. Such a condition is nicely encapsulated by the inverse condition number of a matrix, which can be calculated via the singular value decomposition (SVD) [4]:

$$\gamma^{-1} = \sigma_{min}/\sigma_{max} \tag{5}$$

where σ_{max} is the largest singular value of C and σ_{min} is the smallest, so that a well conditioned matrix scores 1.0 and an ill-conditioned matrix scores close to 0.0.

1.2 HMM Residuals

Ambiguity defined by the condition number of C indicates how well the HMM is likely to cover the dimensionality of the model parameter space. However, it does not measure where rank deficiencies may actually reside. To measure this, we let C_i be the matrix C with row i (r_i), removed. Define $P_{C_i}(r_i)$ as the projection of r_i onto the span of the vectors defined by the row space of C_i. Now define the residual vector:

$$\vec{e}_i = \vec{r}_i - \vec{P}_{C_i}(r_i), \tag{6}$$

and the residual matrix:

$$E = [\vec{e}_1 \vec{e}_2 ... \vec{e}_N]^T. \tag{7}$$

The residual matrix can now 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 is 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:

$$s_j = \sqrt{\sum_i e_{ij}^2}, \tag{8}$$

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_j would also be 1.0. However, since the row space is typically not orthogonal, s_j can be larger than 1.0. Nevertheless, a value of s_j near 1.0 indicates that element j is quite independent of other elements and is therefore important for the HMM. On the other hand, if s_j is close to 0, there are two possibilities. Either element j is highly dependent on the other elements and is therefore not particularly useful to the HMM, or element j is an unlikely state or observation symbol. In the former case, element j can be safely removed from the HMM. In the latter case, the practitioner must decide if element j warrants inclusion. It may, for example, indicate a very unlikely but extremely important event. Note that element j refers to either a state of the HMM (the first N columns of C) or an observation symbol (the next M columns of C).

1.3 The Conditional Information of a HMM

So far we have only considered issues of dimensionality, ambiguity, of the HMM model parameter space. Here we explore this issue of what information a given HMM component structure contributes to the performance of the HMM. In particular, we consider how the A and B parameters contribute to the prediction of state sequences given a model and observations using the conditional information measure from Information Theory[1]. A HMM consists of two components that work in tandem: (1) a classifier which uses the input observation sequence to evidence the state of the HMM (the B matrix); (2) a Markovian component which uses the previous state to evidence the next state (the A matrix) [6].

If the B matrix is unambiguous (for example, orthogonal with an inverse Condition Number 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 best predict the most likely state at time, t. This condition would eliminate the need for the Markovian component (A matrix) by use of a simple Bayesian (ML or MAP) classifier. Conversely, if the model B matrix is quite ambiguous, we may as well dispense with it and just use the Markovian component of the HMM - and its associated Viterbi algorithm. We have used the conditional information as a means of teasing out the contributions of each such component to the solutions for optimal state sequences as follows.

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), $\vec{S_b}$. This latter condition assumes that the predictions at each time period are independent of all others - the assumption for normal HMMs[6]. From the resultant two state sequences, $\vec{S_v}$ and $\vec{S_b}$, respectively, we can then calculate the following quantities:

$$H(v|b) = H(v,b) - H(b) \tag{9}$$

where

$$H(v,b) = -\sum_{i,j}(P(S_v = i, S_b = j) log P(S_v = i, S_b = j)) \tag{10}$$

and

$$H(b) = -\sum_{j}(P(S_b = j) log P(S_b = j)) \tag{11}$$

where $H(v|b)$ 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

$$R(v|b) = H(v) - H(v|b) \tag{12}$$

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 throw new light on the interpretations of past published papers using HMMs as without the type of analysis discussed above, it is unclear as to whether past reported HMMs were ill-conditioned, unnecessary or optimal for a given task. In the following we illustrate how these measures can be used to diagnose and even improve the behaviour of HMMs.

2 Example: Gesture Recognition

We consider a difficult estimation problem to demonstrate the usefulness of the measures - one not uncommon in vision-based gesture recognition. The problem is one of estimating the pose (roll, pitch and yaw) of a hand (in this case, the graphical model of a hand) from it's image. Figure 1 shows an example sequence of the hand. The motion of the model

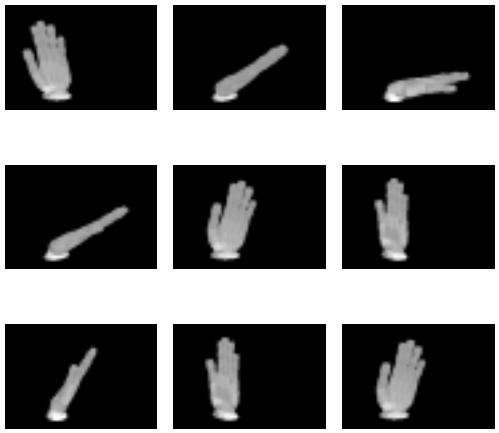

Figure 1. Nine sequential frames of a video sequence used in the deterministic movement condition.

is rigid about the wrist joint. The poses of the hand are quantized so that there are 5 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.

For this example we concentrated on estimating the pitch of the hand only (although roll and yaw also change). To estimate pitch, 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. In this case we have explored performance with 5 states: the need to recover 5 3D poses purely from the image under a number of movement conditions corresponding to a deterministic walk, a random walk, and a set of purely randomly selected poses.

2.1 Case 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 it's next position until the maximum range of motion was reached. The motion then reversed in a backwards fashion. Figure 1 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 these images were generated from 3D CAD models) as well as the 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 to update the model. Initially, we arbitrarily partitioned the observation range into 5 equal symbol bins. This produced the following HMM:

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

$$B = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

$$\pi = [0.06\ 0.13\ 0.62\ 0.13\ 0.06]$$

If we calculate the inverse condition number of the augmented matrix $A|B$, we obtain 0.23 indicating that the HMM is not ideal. The residual sum of the matrix is (Eqn. 8):

$$R^1 = [0.5\ 0.6\ 0.4\ 0.6\ 0.5\ 0.3\ 0.9\ 0.0\ 0.0\ 0.0]. \quad (13)$$

This indicates that only the first 2 observation symbols are useful (recall that the first 5 elements of R^1 refer to the HMM states), and that the rest could be discarded. Clearly R^1_7 (first element is R^1_1) performs better than R^1_6 and the overall prediction accuracy is 63% on the test sequence. In turn, the HMM could be improved by refining symbol R^1_6 and removing symbols R^1_8, R^1_9, R^1_{10}. 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. Consequently, we split R^1_6 into three distinct symbols and removed R^1_8, R^1_9 and R^1_{10} resulting in 4 observation symbols. Re-estimating a new HMM results in:

$$B = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}.$$

This gives an inverse condition number of 0.34 and a residual sum vector of:

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

This HMM results in a perfect prediction accuracy (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 $H(v) = 2.25$, $H(v|b) = 0.75$ and so $R(v|b) = 1.5$ or 33% 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.

2.2 Case 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 backwards 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 a 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:

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

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

$$\pi = [0.20\ 0.16\ 0.30\ 0.14\ 0.20].$$

The inverse condition number 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:

$$R^1 = [0.4\ 0.4\ 0.5\ 0.4\ 0.4\ 0.1\ 0.1\ 0.0\ 0.0\ 0.0].$$

Given the relatively better discriminatory power of the 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 inverse condition number of 0.19, and a residual sum vector of:

$$R^2 = [0.5\ 0.5\ 0.5\ 0.5\ 0.4\ 0.2\ 0.2\ 0.1\ 0.2\ 0.1\ 0.0\ 0.1\ 0.1]$$

and prediction performance of 40% correct.

Continuing this process for three more iterations, resulted in an inverse condition number 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, $H(v) = 2.32$, $H(v|b) = 1.59$ and $R(v|b) = 0.73$ or 69% of the information was contained in the A matrix in predicting the optimal state sequence. This is confirmed since the Bayes classifier performed at 50% correct prediction, significantly less than the complete HMM (75%).

2.3 Case 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 a HMM with an inverse condition number of 0.11 and a residual sum of:

$$R^4 = [0.1\ 0.1\ 0.1\ 0.0\ 0.1\ 0.3\ 0.2\ 0.3\ 0.1\ 0.2\ 0.2\ 0.1\ 0.1\ 0.2\ 0.1\ 0.30.2\ 0.1\ 0.0\ 0.1\ 0.1\ 0.2\ 0.2\ 0.1\ 0.1\ 0.0\ 0.0\ 0.2\ 0.0\ 0.1].$$

So clearly, we have an extremely difficult problem on which the HMM doesn't appear to be doing very well. In fact, the prediction performance is 49% correct. In this case we found that $H(v) = 2.06$, $H(v|b) = 0.56$ and so $R(v|b) = 1.5$ or 27% of the information was contained within the A matrix in predicting the optimal state sequence. This indicates 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.

In all then, we can make two conclusions from these simulations. One, even before running the HMM on data we can already determine how well it can perform on similar data to the training data by the proposed diagnostic tools. Two, past reports on success of HMMs may have nothing to do with HMMs, per se, but rather to the selection of good features or dynamics which can be unambiguously modeled by a simple Markov Chain or Bayesian classifier.

3 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 from deterministic to uniform(random)as:

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

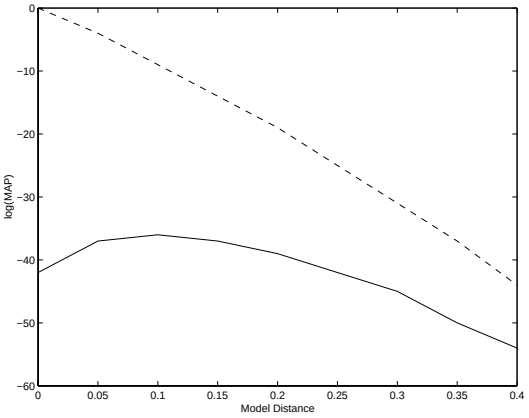

Figure 2. Shows MAP values as a function of average absolute difference between model parameters as a function of the model Condition Numbers and uncertainties. Dashed curve corresponds to the deterministic reference model (Condition Number of 1) and solid curve to the random model (Condition Number of 0)

We selected two HMMs corresponding to: (1) the deterministic case: A and B matrices being the identity with an inverse Condition Number of 1. (2) the random case, with both A and B matrices having values of 0.5 everywhere and an inverse Condition Number 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 a number of 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 Figure 2. Here model distance was simply defined as the total absolute distance between entries in the model C matrices (Eqn. 4). 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 Discussion

Here 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 sequences 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.

Furthermore these tolls provide the practitioner with methods for improving the model. 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.

Finally, we have also observed that uses of the Viterbi algorithm and MAP score is by no means reliable for pattern recognition - that is, model identification, as the model departs significantly from a deterministic one. That is, for the extreme case of a random model model discrimination is near impossible until the comparison model parameters are on average 50% different from the reference model.

Together, this analysis clearly demonstrates the need for HMM components analysis and parameter documentation in future uses of HMMs in Computer Vision studies before any scientific conclusions can be made about claims that a HMM model is appropriate for solving a given spatio-temporal pattern recognition problem.

References

[1] R. Ash. *Information Theory*. Interscience Publishers, 1995.

[2] T. Caelli, A. McCabe, and G. Binsted. On learning the shape of complex actions. In *International Workshop on Visual Form: IWVF'2001, Lecture Notes in Computer Science, LNCS 2059, Springer*, volume 15, pages 197–221. Morgan Kaufmann, San Mateo, CA, 2001.

[3] Z. Ghahramani and M. Jordon. Factorial hidden markov models. *Machine Learning*, 29:245–275, 1997.

[4] G. H. Golub and C. F. V. Loan. *Matrix Computations*, chapter 2.7, pages 79–81. The Johns Hopkins University Press, 2nd edition, 1989.

[5] J. Norris. *Markov Chains*. Cambridge University Press, Cambridge, England, 1997.

[6] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.