Department of Computer Science,
University of Otago

Technical Report OUCS-2002-05

Diagnostic Tools for Evaluating HMM Components

Authors:
Brendan McCane
Department of Computer Science, University of Otago

Terry Caelli
Department of Computer Science, University of Alberta

Status: submitted to
IEEE Transactions on Pattern Analysis and Machine Intelligence

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/trseries/

Diagnostic tools for evaluating HMM components

Brendan McCane[*]

Department of Computer Science, University of Otago

Dunedin, New Zealand, email: mccane@cs.otago.ac.nz

Terry Caelli

Department of Computer Science, University of Alberta

Edmonton, Canada

June 21, 2002

Abstract

Although there are known algorithms for predicting observation and state sequences from HMM models there is little discussion on how to determine the contributions of the different types of HMM parameters to such predictions and consequently, temporal pattern recognition. In this note we discuss and compare a number of objective measures that provide insight into HMM performance in these terms.

Keywords: HMM, evaluation.

1 Introduction

Over the past 20 years hidden Markov models (HMMs) have become a popular method for encoding, recognizing and predicting sequential patterns of univariate and multivariate observation data. The Viterbi algorithm has been

[*] corresponding author

the most popular method for predicting optimal state sequences and it's associated maximum posterior probability (MAP) score is typically used for temporal pattern recognition and classification. For predicting observation sequences, Monte Carlo sampling is typically used [4, 2]. However, such methods and measures have not been analyzed in relation to the specific model parameters and, in particular, as to how each parameter set actually contributes to performance. Accordingly, here we have investigate three discrete HMM diagnostic tools to answer this question: the HMM Condition Number, Residual Sum Vector, and Conditional Information content.

We follow the HMM nomenclature of Rabiner [4] where there exists a model, a sequence of state symbols, S_i, and a sequence of observation symbols, O_j. In most cases we need to estimate the state sequences from one sequence of observation symbols, given a model. The model, 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 and

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

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

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

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

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

In this note we investigate how to assess the contributions of the individual HMM components to the prediction of observations or optimal state sequences. This issue bears upon how we interpret HMMs and how they can or cannot informatively versus predictably represent what is observed. For example, imagine a system that consists of two coins. One coin has a head on both sides, and the other coin has a tail on both sides. At any trial, one of the coins is chosen randomly and flipped. The result of the flip is then recorded - either heads or tails. We can construct a HMM that models this system as:

$$A = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \qquad B = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \tag{4}$$

Although this HMM is an informative statistic of what is observed (over large sample sizes) the A matrix has no predictive power. On the other hand, given an observation it is clear that the state-dependent observation matrix (B) can fully predict the state at any time - in this case, demonstrating the importance of the non-Markovian model component. In this vein, then, we discuss three simple, but useful, diagnostic tools for determining just how these HMMs components contribute to the prediction of observations and state sequences.

1.1 Condition Number of a HMM

Experienced users of HMMs probably already 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:

$$C = A|B. \tag{5}$$

Accordingly, we would like the rows of C to be as linearly independent as possible. The more independent, the less “ambiguous” the HMM. Such a condition is nicely encapsulated by the inverse condition number of a matrix, which can be calculated via the singular value decomposition (SVD) [3]:

$$\gamma^{-1} = \sigma_{min}/\sigma_{max} \tag{6}$$

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

The condition number of C indicates how well the HMM is likely to cover the dimensionality of the model parameter space, excluding the initial state distribution vector, π. However, it does not measure where rank deficiencies may 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:

$$e_i = r_i - P_{C_i}(r_i), \tag{7}$$

and the residual matrix:

$$E = [\vec{e}_1 \vec{e}_2 ... \vec{e}_N]^T \tag{8}$$

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 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:

$$s_j = \sqrt{\sum_i e_{ij}^2}, \tag{9}$$

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 symbol j is quite independent of other symbols and is therefore important for the HMM. On the other hand, if s_j is close to 0, there are two possibilities. Either symbol j is highly dependent on the other symbols and is therefore not particularly useful to the HMM, or symbol j is an unlikely state or observation symbol. In the former case, symbol j can be safely removed from the HMM. In the latter case, the practitioner must decide if symbol j warrants inclusion. It may, for example, indicate a very unlikely but extremely important event. Note that symbol 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 of the HMM model parameter space and, in Section 2 we will explore the effects of this on HMM performance. We now consider how the A and B parameters contribute to the prediction of state sequences given a model and observations. The diagnostic tool is based on the conditional information measure from Information Theory[1] in the following way. A HMM consists of two components which work in tandem: a classifier or clusterer which uses the input observation sequence to evidence the state of the HMM (the B matrix); and a Markovian component which uses the previous state to evidence the next state (the A matrix) [4].

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 this "classification" part and just use the Markovian component of the HMM - and its associated Viterbi algorithm. We propose the use of conditional information as a means of teasing 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), $\vec{S_b}$. This latter condition assumes that the predictions at each time period are independent of all others. Given the resultant two state sequences, $\vec{S_v}$ and $\vec{S_b}$, respectively, we can calculate the following quantity:

$$H(v|b) = H(v,b) - H(b) \tag{10}$$

where

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

and

$$H(b) = -\sum_{j}(P(S_b = j) log P(S_b = j)). \tag{12}$$

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

$$R(v|b) = H(v) - H(v|b) \tag{13}$$

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, hopefully, 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 Synthetic and Real Examples

2.1 Synthetic Case: Experimental Setup

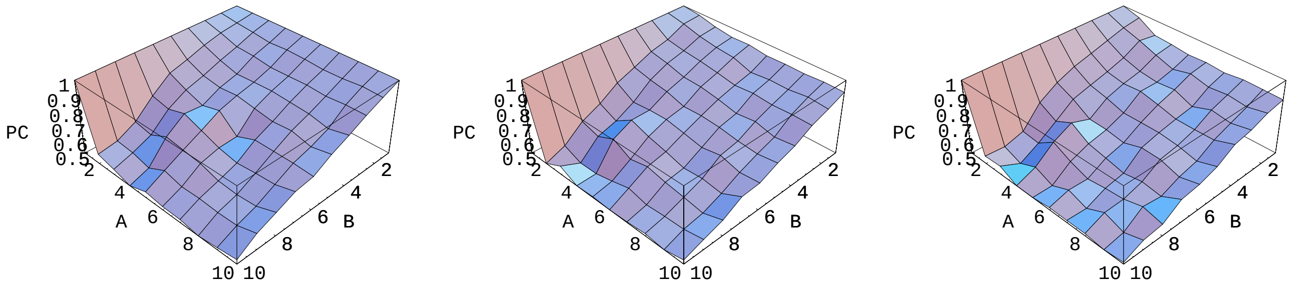
To demonstrate the usefulness of the above diagnostic tools, we have analyzed a very simple class of HMMs having two states and two observation symbols with systematically varying probabilities. From these HMMs we generated test sequences using Monte Carlo model sampling from which we could perform the proposed measurements. We generated the HMMs by varying both the A and B matrices from deterministic to random, independently. We used 10 steps between deterministic and random for each of A and B, as:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix} \quad \cdots \quad \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \tag{14}$$

resulting in a total of 100 HMMs.

2.2 Results

In the first experiment we tested how well we can consistently estimate state sequences generated from each of the above underlying HMMs in the presence of noisy observation sequences. We expected that it would be more difficult to do so for random HMMs than for more deterministic ones. To enable such analyses, for each HMM we generated two example observation sequences of length 1000 using Monte Carlo sampling of the model parameters, and in accord with the model definition. Both observation sequences were then perturbed with varying levels of random noise by randomly selecting $x\%$ of the observations of a given sequence, and randomly re-assigning that observation symbol. Given these two noisy observation sequences, we estimate a new HMM model using a moving window (partial fractions) co-occurrence method and the Baum-Welch algorithm on one of the sequences [4]. We estimate a new HMM to simulate the (normal) situation where we don't know the true underlying model. We then ran the Viterbi algorithm on the second sequence to produce the most likely state sequence.



(a) Noise=0%

(b) Noise=15%

(c) Noise=25%

Figure 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).

This state sequence was compared with a state sequence generated using the Viterbi algorithm and an observation sequence generated (without noise) from the original model using a percentage correct (PC) measure. This method provides a way of determining both the robustness of the model and the degree to which this robustness can be indexed by the proposed measures. The choice of two observation sequences, one for re-estimation and one for prediction was to keep estimation and prediction as two different processes and determined from two independent samples from the same model.

The results are shown in Figure 1 where both axes vary from deterministic (=1) to random (=10). As can be seen from the figure, and perhaps unsurprisingly, as both the A and B matrices become more random, the ability to estimate the optimal state sequence of the original HMM degrades. In fact, a PC of 50% 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.

Figure 2 shows how the inverse condition number 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 inverse condition number 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.

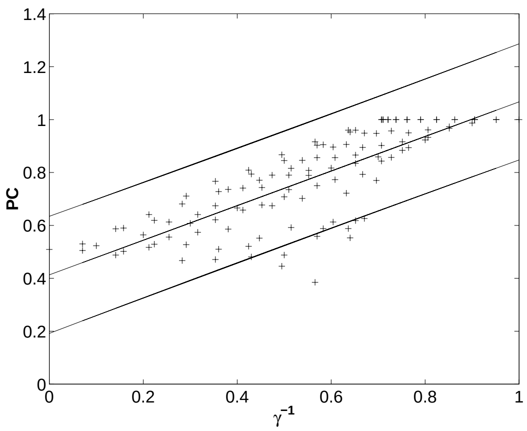


Figure 2: Shows the relationship between inverse condition number (γ^{-1}) and percentage correct (PC) prediction of observations. The raw data is plotted along with the least squares regression line and the upper and lower 95% confidence interval lines.

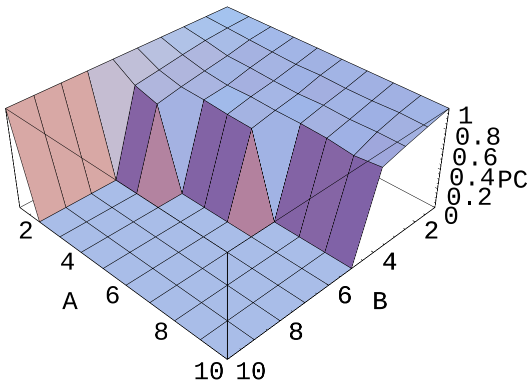
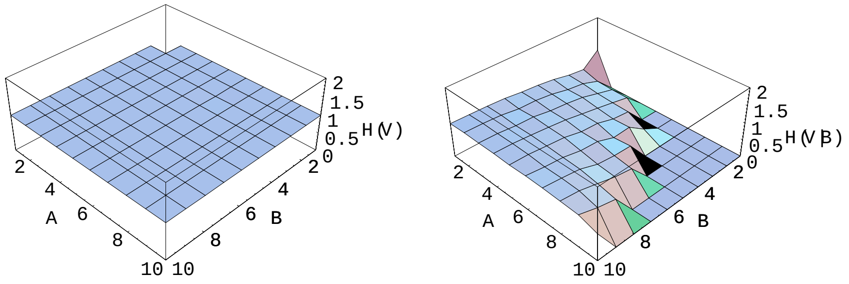


Figure 3: Results for estimating state sequences with a threshold of 80%. We have varied each of the A matrix and B matrix from deterministic (axis value = 1) to random (axis value = 10).

From Figure 1 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 Figure 1(a) to observe what range of parameters satisfies this criterion. The result is shown in Figure 3. This figure clearly highlights that the B matrix is the most important factor in the success of a HMM. That is, A is required to be near deterministic to affect the performance over the B matrix - an interesting result given the simple Bayesian classifier interpretation of the B matrix.

We have performed a similar experiment to those discussed using the $H(v)$ and $H(v|b)$ conditional information values as the dependent variables. These results are shown in Figure 4. As can be seen the entropy of the Viterbi



(a) H(v)

(b) H(v|b)

Figure 4: Shows, as a function the different model parameters: $H(v)$: information in the Viterbi solution; $H(v|b)$: the information contained in the the Viterbi solution not contained in the Bayes classifier.

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. Figure 4(b) clearly indicates that a 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 Figure 3 the $H(v|b)$ 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 a HMM-based method for recognizing hand movements contribute to the performance prediction.

3 An Illustrative Real Example

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 5 shows an example sequence of the hand. The motion of the model is rigid about the wrist joint. The poses of the hand are quantized so that there are 5 possible

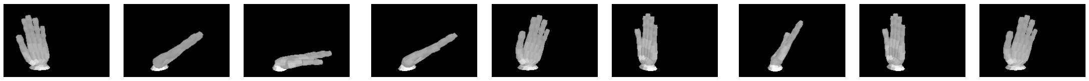


Figure 5: Nine sequential frames of a video sequence used in the deterministic movement condition.

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.

3.1 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 5 shows nine frames of the sequence.

We generated two 1000 length sequences and store both the ground truth data (i.e. the actual pitch) 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. Initially, we arbitrarily partitioned the observation range into 5 equal bins. This produced the following HMM:

$$A = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 & 0.5 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix} \qquad B = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{15}$$

$$\pi = [0.06\ 0.13\ 0.62\ 0.13\ 0.06] \tag{16}$$

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:

$$R^1 = [0.5\ 0.6\ 0.4\ 0.6\ 0.5\ 0.3\ 0.9\ 0.0\ 0.0\ 0.0]. \tag{17}$$

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 symbol is R^1_1) performs better than R^1_6 and the overall prediction accuracy is 63%. 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:

$$B = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}. \tag{18}$$

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

$$R^2 = [0.4\ 0.7\ 0.5\ 0.7\ 0.6\ 0.4\ 0.0\ 0.8\ 0.9]. \tag{19}$$

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 $H(v) = 2.25$, $H(v|b) = 0.75$ 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.

3.2 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:

$$A = \begin{bmatrix} 0.46 & 0.54 & 0.00 & 0.00 & 0.00 \\ 0.45 & 0.00 & 0.55 & 0.00 & 0.00 \\ 0.00 & 0.53 & 0.00 & 0.47 & 0.00 \\ 0.00 & 0.00 & 0.46 & 0.00 & 0.54 \\ 0.00 & 0.00 & 0.00 & 0.55 & 0.45 \end{bmatrix} \quad B = \begin{bmatrix} 0.74 & 0.26 & 0.0 & 0.0 & 0.0 \\ 0.90 & 0.10 & 0.0 & 0.0 & 0.0 \\ 0.67 & 0.33 & 0.0 & 0.0 & 0.0 \\ 0.89 & 0.05 & 0.04 & 0.03 & 0.0 \\ 0.86 & 0.14 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{20}$$

$$\pi = [0.20\ 0.16\ 0.30\ 0.14\ 0.20]. \tag{21}$$

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:

$$R^1 = [0.4\ 0.4\ 0.5\ 0.4\ 0.4\ 0.1\ 0.1\ 0.0\ 0.0\ 0.0]. \tag{22}$$

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:

$$R^2 = [0.5\ 0.5\ 0.5\ 0.5\ 0.4\ 0.2\ 0.2\ 0.1\ 0.2\ 0.1\ 0.0\ 0.1\ 0.1] \tag{23}$$

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 ML classifier performed at 50% correct prediction, significantly less than the complete HMM (75%).

3.3 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:

$$R^4 = [0.1\ 0.1\ 0.1\ 0.0\ 0.1\ 0.3\ 0.2\ 0.3\ 0.1\ 0.2\ 0.2\ 0.1\ 0.1\ 0.2\ 0.1\ 0.30.2\ 0.1\ 0.0\ 0.1\ 0.1\ 0.2\ 0.2\ 0.1\ 0.1\ 0.0\ 0.0\ 0.2\ 0.0\ 0.1] \tag{24}$$

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. 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.

4 Discussion

In this note, 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, as demonstrated in Section 3, the practitioner can follow a methodical routine to improve a 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 quantisation as in the examples above), and to extend the analysis to coupled HMMs.

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.

References

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

[2] H. Bunke and T. Caelli (Eds). Special edition on on hidden markov models in vision. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1), 2001.

[3] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, chapter 2.7, pages 79–81. The Johns Hopkins University Press, 2nd edition, 1989.

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