Effect of Initial HMM Choices in Multiple Sequence Training for Gesture Recognition

Nianjun Liu, Richard I.A. Davis, Brian C. Lovell and Peter J. Kootsookos
Intelligent Real-Time Imaging and Sensing (IRIS) Group
School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane, Australia 4072.
Email: {nianjunl, riadavis, lovell, kootsoop}@itee.uq.edu.au

Abstract

We present several ways to initialize and train Hidden Markov Models (HMMs) for gesture recognition. These include using a single initial model for training (re-estimation), multiple random initial models, and initial models directly computed from physical considerations. Each of the initial models is trained on multiple observation sequences using both Baum-Welch and the Viterbi Path Counting algorithm on three different model structures: Fully Connected (or ergodic), Left-Right, and Left-Right Banded. After performing many recognition trials on our video database of 780 letter gestures, results show that a) the simpler the structure is, the less the effect of the initial model, b) the direct computation method for designing the initial model is effective and provides insight into HMM learning, and c) Viterbi Path Counting performs best overall and depends much less on the initial model than does Baum-Welch training.

Keywords: Hidden Markov Model(HMM), Baum-Welch learning, Viterbi Path Counting, Discrete observation sequences, Segmentation.

1 Introduction

Hidden Markov Models (HMMs) have been widely used with considerable success in speech and handwriting recognition during recent decades[1, 2, 3]. More recently they have been successfully applied to problems in computer vision and pattern recognition. Schlenzig and Hunter (1994) designed the recursive identification of gesture inputs using Hidden Markov Models to recognize three gestures [6]. Later, Starner (1998) used Hidden Markov Models for visual recognition of American Sign Language (ASL) [4]. In a related project, Lee and Kim (1999) designed an HMM-based threshold model approach for recognition of ten gestures to control PowerPoint slides [5].

In earlier work, Davis and Lovell (2002) studied the effects of varying weighting factors in learning from multiple observation sequences [7]. Later, Davis and Lovell (2003) compared multiple-sequence Baum-Welch [1], Ensemble methods [9], and Viterbi Path Counting methods [10, 8] on synthetic and real data. This study determined that Viterbi Path Counting was the best and fastest method. However the problem of choosing the initial model for the training (re-estimation) process was left as an open problem and is the topic of this paper.

Here we investigate three different ways of generating the initial models used in training: 1) Single Random initial models, 2) Multiple Random initial models, and 3) Directly Computed initial models based on physical insight into the problem at hand.

We apply three kinds of Baum-Welch training algorithms: equal weight Baum-Welch method, the Ensemble method and the Viterbi Path Counting method to evaluate the initial model effects. The model structures used are Fully Connected (FC), Left-Right (LR), and Left-Right Banded (LRB).

In trials of a gesture recognition system for a set of 26 single-stroke letter gestures (described in more detail in [11]), the judicious choice of initial models provided a substantial improvement on the recognition rate in FC models. The results also show that 1) the simpler the HMM structure, the less effect the initial model has, 2) that Viterbi Path Counting performs better and is less sensitive to initial model choice than Baum-Welch. This is particularly true on LRB models where VPC attained the overall best performance of any method.

2 HMM Initial Model Design

A Hidden Markov Model consists of N states, each of which is associated with a set of M possible observations. It includes the initial parameter π, a transition matrix A and an observation matrix B, and can be specified by the HMM model parameter set $\lambda = (A, B, \pi)$.

2.1 Static Initial Parameter

The use of a single initial model is common in Baum-Welch training. When we train HMMs on multiple observation sequences, we only set up the initial model once, and all observation sequences are trained on the same initial model. The model trained using this method is highly dependent on the choice of initial model, and the performance of the trained models vary greatly. The reason is that Baum-Welch is only able to find a local maximum, not a global maximum. Thus if the initial model is located near the global maximum, the trained model works well, but not otherwise.

2.2 Multiple Random Initial Models

To overcome the problem of the single initial model choice, in this method we generate multiple initial models, with the number being the same as the number of observation sequences. At the same time, we make them evenly distributed in the space. We expect that this group of initial models should include several points near the global maximum. We design the ensemble method for training the HMMs so that each observation sequence is trained on its own initial model.

2.3 Directly Computed Models

The main motivation for considering directly computed HMMs was to gain a deeper insight into the way HMMs learn gesture patterns and to determine more reasonable initial models than random starts.

Here we present a way of pre-computing the initial model directly from the training data set. We derive the direct computation method by analyzing the LRB structure (figure 2). The LRB model structure allows only self-transitions ($S_n \to S_n$) and transitions to the next indexed state ($S_n \to S_{n+1}$), until finally reaching the terminal state. We therefore attempt to segment the gesture (i.e., segment the observation sequence) according to the expected duration time of each state. The simplest approach is to evenly segment the total duration of the gesture according to the number of states, and then determine the A matrix from the time duration equation (2). The A matrix can be computed using the "true" state path and the duration equation. The observation sequence overlaid on the state path can be treated as Gaussian positional noise, so the B matrix can be directly computed by fitting a Gaussian mixture to the histogram on the real data set and solved through the corresponding equations. An example of a directly computed initial model is shown in figure 1.

$$A = \begin{bmatrix} 0.83 & 0.17 & 0 & 0 \\ 0 & 0.83 & 0.17 & 0 \\ 0 & 0 & 0.83 & 0.17 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

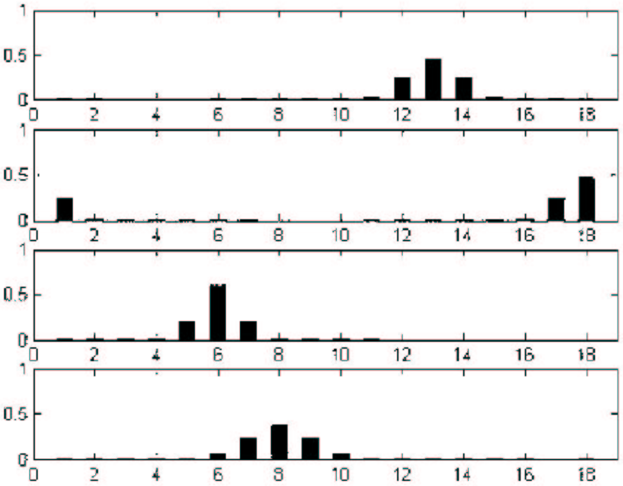

Figure 1: Max-Value Gaussian Distribution showing A values and B histograms for 4 states.

2.3.1 A Matrix Computation

The A matrix is computed using the state duration equation (duration time in a state, conditioned on starting in that state):

$$d_i = \frac{1}{1 - a_{i,i}} \tag{2}$$

where d_i is the state duration time and, and $a_{i,i}$ is the A-matrix parameter. The observation sequence (T) is segmented evenly and the duration time is the same for each state. For example, if 3 states are used, the observation sequence length (T) is 24, and the duration time is 24/3=8; if $d = 8$, then $a_{i,i} = 0.875$, and because the row sum is 1, then the other value is 0.125. In a similar way, $a_{3,3} = 1$; so the A matrix is fully calculated. The initial A matrices for FC and LR structures are generated randomly, and we choose the largest two elements of each each row i and place these in positions $a_{i,i}$ and $a_{i,(i+1)}$. This method forces the the randomly generated matrix to be closer to the LRB form which is known to produce good recogntion rates.

2.3.2 B Matrix Computation

We calculate a Gaussian mixture distribution based on the observation histogram distributions to compute the B matrix.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{3}$$

$$\sigma = \frac{1}{f(0)\sqrt{2\pi}} \tag{4}$$

where μ,σ are the Gaussian mean and standard deviation parameters respectively for the distribution f.

We first plot the histograms of observations in the training data set. There are 18 possible values for the observation symbols, the observation sequence length or period (T) is 24 (the videos consist of 25 frames so there are 24 angle symbols for each interframe interval), and the training set comprises 20 observation sequences ($T \times 20$). We segment the training set evenly according to the number of states (N). Each segmentat has duration T/N, and we histogram the observations to estimate the pdf. For example, if there are 3 states and the period T is 24, state 1 corresponds to the first 8 observations, state 2 to the second 8, and state 3 to the third 8, for all 20 observation sequences. Histogramming the observations for each state separately, we find that the histograms have several peaks. We choose no more than three maximum peaks and fit Gaussian Distributions (equation) around each peak. Finally, we normalize the distribution to sum to 1 so it can be used as a row of the B matrix.

3 Two Types of Baum-Welch and Viterbi Path Counting

We used the traditional Baum Welch [1] and Viterbi Path Counting [8] algorithms to train the HMM models.

3.1 Three Classes of Model Structure

There are three types of model structures shown in figure 2. In an FC HMM, every state of the model can be reached from every other state of the model. The characteristic property of LR HMMs is that no transitions are allowed to states whose indices are lower than the current state. The third special model structure is the LRB model. This has a transition structure consisting of a single linear chain containing only self-transitions and transitions from elements to the following element in the linear chain.

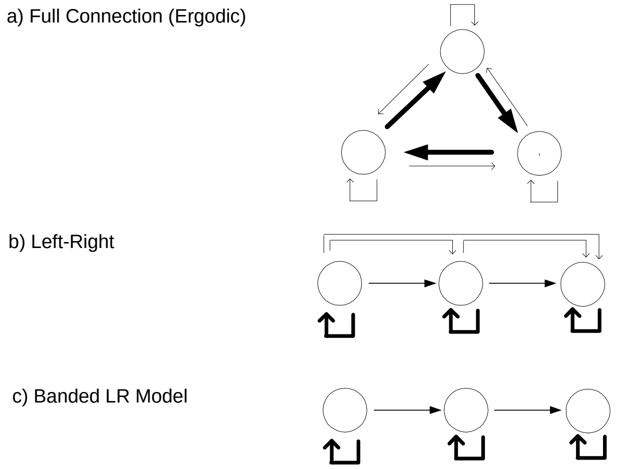

Figure 2: Three types of model structure

3.2 Equal Weight Baum-Welch

In our system, each gesture has multiple observation sequences (Numseq=K). The most common method is to use the multi-sequence training algorithm proposed by Rabiner and Juang [1]. This uses the K observation sequences at each stage of the Baum-Welch re-estimation process to iteratively update a single HMM parameter set. The re-estimation formulae for this type of iterative method are as follows:

$$\overline{a}_{ij} = \frac{\sum_k W_k \sum_{t=1}^{T_k} \alpha_i^k a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^{(k)}(j)}{\sum_k W_k \sum_{t=1}^{T_k} a_t^k(i) \beta_t^k(i)} \tag{5}$$

$$\overline{b}_{ij} = \frac{\sum_k W_k \sum_{O_t(k)=v_j} \alpha_t^k(i) \beta_t^k(i)}{\sum_k W_k \sum_{t=1}^{T_k} \alpha_t^k(i) \beta_t^k(i)} \tag{6}$$

where $W_k = 1/P_k, k \in [1, K]$ is the inverse of the probability of the current model estimate generating training sequence k, evaluated using the forward algorithm. In our system, we set all $W_k = 1$, and achieve similar results to those obtained by setting $W_k = 1/P_k$ (see [7] for examples of varying these weights on synthetic data). O_k is the observation symbol at time t emitted by sequence k. The forward and backward algorithms define $\alpha_t^k(i)$ and $\beta_t^k(i)$ for sequence k, time t and state i respectively.

The Baum-Welch algorithm is an "iterative update" algorithm which re-estimates parameters of a given Hidden Markov Model to produce a new model which has a higher probability of generating the given observation sequence. This re-estimation procedure is continued until no more significant improvement in probability can be obtained and the local maximum is thus found. However, the training results depend greatly on the choice of the initial model.

3.3 Ensemble Method

The approach described here is a special case of the method suggested by Mackay [9] where an ensemble of models is trained. It has been studied on synthetic data in [8]. This combines ML methods in a Bayesian merging step to maximize posterior probability. In this adaptation of the method, one model is estimated for each of the K observation sequences. Other approaches are possible but are not considered here. This enables the formation of K independent model estimates from the training sequences. From these estimates, the next step is to examine the efficacy of combining the independent parameter estimates using a range of simple averaging techniques of the following form:

$$\overline{a}_{ij} = \frac{\sum_k W_k a_{ij}^{(k)}}{\sum_k W_k} \tag{7}$$

$$\overline{b}_{ij} = \frac{\sum_k W_k b_{ik}^{(k)}}{\sum_k W_k} \tag{8}$$

$$\overline{\pi}_i = \frac{\sum_k W_k \pi_i^{(k)}}{\sum_k W_k} \tag{9}$$

where W_k is the weighting factor for each sequence and $\lambda^{(k)} = (A^{(k)}, B^{(k)}, \pi^{(k)})$. Refer to [7] for a range of synthetic data results. The quality of all model estimates is judged by the probability of that model generating an unseen set of test sequences from the same source as the K training sequences as described below. The intention is to improve on Rabiner and Juang's method described above using a weighted combination of an ensemble of learnt models to avoid local minima traps as much as possible. Although each sequence is matched to the structure of each model, structure is not incorporated in the averaging step itself. It therefore places a limited reliance upon structure information. In this experiment only uniform weights are employed.

3.4 Viterbi Path Counting Algorithm

The Viterbi Path Counting (VPC) training method is discussed by Davis and Lovell [8]. The method is to use the Viterbi algorithm to find the most likely path for a given sequence, and to modify the model parameters along that path by maintaining matrices of integers corresponding to Viterbi Path statistics for π, A and B. It was thought that this would provide a simple and reliable way of ensuring the correct relative importance of evidence between a new sequence and the existing model structure, thus achieving good learning for both single and multiple sequence HMM training.

There are many possibilities for the initialization of the VPC algorithm, including random and uniform count matrices, which can be scaled to any arbitrary amount. In this paper we only investigate uniform counts with a scaling factor of 5; i.e. each counter for each of A, B, π starts with a value of 5.

4 Experiments on the Gesture Recognition System and Discussion

In order to evaluate the initial model performance, we used a video gesture recognition system which can recognize 26 gestures corresponding to the letters of the alphabet. There were 20 training samples and 10 test samples for each distinct gesture; so there are 760 gesture videos in the database in total. After trajectory estimation and smoothing, the angle of movement of the centre of the hand was discretized into 1 to 18 directions over the 25 frames of video to form the discrete observation sequences. Figure 3 illustrates several letters and shows how they were recorded.

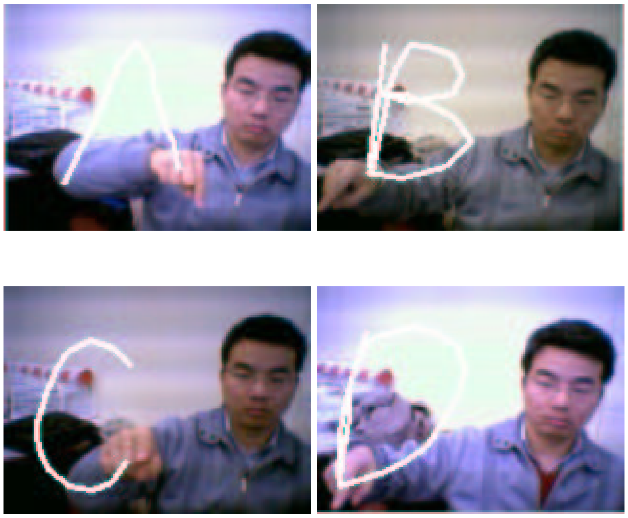

Figure 3: Hand gestures for A, B, C and D

Since the length of the observation sequence (T) is 24, this is easy to divide evenly into N=3, 4, 5, or 8 segments. When N is 5,7,9 and 10, we distributed the remainder evenly between the last segments. Thus for N=5, our segment lengths were 4, 5, 5, 5, and 5.

We apply the traditional Baum-Welch [1] and the Viterbi Path Counting [8] algorithms to train HMMs over FC, LR, and LRB structures, with the number of states ranging from 3 to 10.

Figure 4 shows the recognition rates on the real gesture data set with 26 gestures. Each value is an average over 10 test samples for all 26 gestures. The column of means shows the average over all numbers of states. The spread around this mean shows the sensitivity to model size. After analyzing the table of recognition rates, we draw the following conclusions:

1. For the FC structure, whether using Baum- Welch or VPC, the initial model choices have a dramatic effect

Baum Welch	State=3	4	5	6	7	8	9	10	Mean
1. FC/S	66.54	80.00	75.20	75.60	77.60	76.80	77.60	76.00	75.67
2. LR/S	92.31	84.80	81.20	84.80	86.40	86.00	85.60	81.60	85.34
3. LRB/S	92.15	85.38	90.77	85.77	89.62	89.62	90.00	88.46	88.97
4. FC/M	76.15	71.92	72.69	74.23	70.00	71.54	75.38	75.38	73.41
5. LR/M	90.77	92.69	93.08	94.23	93.85	91.54	90.38	90.38	92.12
6. LRB/M	90.00	95.77	94.62	89.23	86.15	75.77	56.54	19.23	75.91
7. FC/P	90.00	91.15	90.38	89.23	90.38	91.54	93.08	92.69	91.06
8. LR/P	89.62	90.00	90.38	89.62	88.46	89.23	90.00	90.38	89.71
9. LRB/P	90.00	90.00	89.62	90.00	89.23	90.00	89.23	90.00	89.76
Difference	State=3	4	5	6	7	8	9	10	Mean
10 FC/M-FC/S	12.6	-11.2	-3.5	-1.8	-10.9	-7.4	-2.9	-0.8	-3.1
11 LB/M-LB/S	-1.7	8.5	12.8	10.0	7.9	6.1	5.3	9.7	7.4
12 LRB/M-LRB/S	-2.4	10.8	4.1	3.9	-4.0	-18.3	-59.2	-360.0	-17.2
13 FC/P-FC/S	26.1	12.2	16.8	15.3	14.1	16.1	16.6	18.0	16.9
14 LR/P-LR/S	-3.0	5.8	10.2	5.4	2.3	3.6	4.9	9.7	4.9
15 LRB/P-LRB/S	-2.4	5.1	-1.3	4.7	-0.4	0.4	-0.9	1.7	0.9
16 FC/P-FC/M	15.4	21.1	19.6	16.8	22.5	21.8	19.0	18.7	19.4
17 LR/P-LR/M	-1.3	-3.0	-3.0	-5.2	-6.1	-2.6	-0.4	0.0	-2.7
18 LRB/P-LRB/M	0.0	-6.4	-5.6	0.9	3.5	15.8	36.6	78.6	15.4
VPC	State=3	4	5	6	7	8	9	10	Mean
19. FC/SU	63.85	53.20	59.60	55.20	45.60	44.40	49.20	43.20	51.78
20. LR/SU	91.15	91.20	91.20	90.40	91.20	90.40	90.40	90.00	90.74
21. LRB/SU	93.08	90.38	95.00	93.85	94.23	94.23	94.62	95.00	93.80
22. FC/P	70.00	71.15	77.31	78.08	77.31	78.46	77.31	75.00	75.58
23. LR/P	90.00	90.77	90.38	91.15	91.54	90.38	90.00	89.62	90.48
24. LRB/P	92.31	90.38	90.38	91.54	90.77	90.00	89.62	90.38	90.67
Difference	State=3	4	5	6	7	8	9	10	Mean
25. FC/P-FC/S	8.79	25.23	22.91	29.30	41.02	43.41	36.36	42.40	31.49
26. LR/P-LR/S	-1.28	-0.47	-0.91	0.82	0.37	-0.02	-0.44	-0.42	-0.29
27. LRB/P-LRB/S	-0.83	0.00	-5.11	-2.52	-3.81	-4.70	-5.58	-5.11	-3.45

Figure 4: The recognition rates by three kinds of initial models. We use EM (Ensemble Method), EW (equal weight Baum Welch), VPC (Viterbi Path Counting), IM (Initial Model), $/S$ (Single Initial Model using EW), $/SU$ (Single Uniform Initial Model under VPC with A, B, π counters set to 5), $/M$ (Multiple Initial Models using EM), $/P$ (Directly pre-computed IM using EW). FC (Full Connection), LR (Left-Right), LRB (Left-Right Banded) are the three structures. BW and VPC are the training algorithms, and $LB/P - LB/S$ (and other similar terms) denotes the difference of the recognition rate between the pre-computed initial model with the LR structure and the single random initial model with LR structure.

on the recognition rate (rows 1,4,7,19,22). Comparing the recognition rates for a Single initial model and Pre-computed initial models, we see a great improvement. The average improvement for BW is 16.9%, and the improvement for VPC is 31.5%. The effect from VPC is stronger than from BW. FC has too many possible ways to run, so the single initial model tends to reach a local maximum, not the global maximum. If we set the pre-computed model as the initial model, there will be a greater chance of reaching the global optimum.

2. The effects of improving the initial parameters in the LR models(rows 2,5,8,20,23) are less apparent than for FC models. The average improvement is only 7.4% and -0.3% for BW and VPC respectively. For the LR-Banded structure (rows 3,6,9,21,24), the mean improvements are only 0.9% and -3.45% respectively. In terms of the level of dependence on the initial model choice, FC is high, LR is medium, and LRB is low. That means that simpler models are less dependent on initial model selection. It seems that the simpler the structure is, the more chance there is of reaching the global optimum.

3. For VPC (rows 19-24), the effect of the initial model choice for both LR and LR-Banded models is quite small. Some states show a negative improvement on LRB. The reason is that LRB is the simplest structure, so it is easy to reach the global optimum. Uniform initial counts worked well in this trial; future work will expand the scope of this survey. The pre-defined IM only provides a general direction for learning, so some

random or flat generating IM may be closer to the correct IM. The results suggest that both LR and LRB structures for VPC are less dependent on the IM choice than for Baum-Welch. It is also support for the viewpoint that VPC is also not as dependent on the initial model selection. VPC produced the best average score of 93.7% on LRB models, and achieved the highest peak of 97% in a separate trial using larger numbers of states (13).

4. The multiple randomly generated initial models (rows 4,5,6) produced good results under Baum-Welch with LR models, outperforming VPC on LR models but not on LRB models. This further supports the idea that VPC is best-suited in cases where the model structure is known. The performance of the ensemble method was quite good on LR models, suggesting that it may be the best method in cases where the structure is not well known.

5 Conclusions

We have estimated the performance of HMMs trained from three kinds of initial models on 26 letter gesture input systems. Three training algorithms and three model structures were adopted during the training process. From the results, we found that the directly pre-computed model method is a novel and reliable way of designing the initial model and has a good chance of reaching the global optimum. It performs much better than the random initial models using BW, especially in the FC model structure. The second conclusion is that because the model structure is simpler, it has less dependence on the initial model. The third is that more restricted models using Viterbi Path Counting perform better and the dependence on the initial model is relatively less than for models created using Baum-Welch. In fact VPC using pre-computed models is not as good as VPC with uniform initial models. In cases where the best model structure is not well known, Ensemble Baum-Welch seems to be the best method. Viterbi Path Counting seems to be the best method when the ideal model structure is known. For our gesture recognition system the best structure was the Banded Left-Right model. The application of Baum-Welch to models initialized using the Viterbi Path Counting algorithm would improve performance further and will be the subject of future research.

References

[1] L.R. Rabiner, and B.H. Juang, Fundamentals of Speech Recognition New Jersy Prentice Hall,1993.

[2] A. Kundu, Yang He and P. Bahl, ”Recognition of Handwritten Words: First and Second Order Hidden Markov Model Based Approach, ” Pattern Recognition, vol. 22, no. 3, p. 283, 1989.

[3] H. Bunke, T. Caelli, “Hidden Markov Models - Applications in Computer Vision”, World Scientific Publishing Co, 2001.

[4] T.Starner and A.Pentland, Real-time American Sign Language Recognition”, IEEE Trans, On Pattern Analysis and Machine Intelligence, Vol 20.pp 1371-1375, Dec.1998.

[5] Hyeon-Kyu Lee and Jin H. Kim, An HMM-Based Threshold Model Approach for Gesture Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence October 1999 (Vol. 21, No. 10). pp. 961-973

[6] J. Schlenzig, E. Hunter, and R. Jain. Recursive identification of gesture inputs using hidden markov models. In WACV94, pages 187–194, 1994.

[7] R. I. A. Davis, and B. C. Lovell, and T. Caelli (2002) *Improved Estimation of Hidden Markov Model Parameters from Multiple Observation Sequences*. Proceedings of the International Conference on Pattern Recognition (ICPR2002), August 11-14 II, pp. 168-171, Quebec City, Canada.

[8] R. I. A. Davis and B. C. Lovell, “Comparing and Evaluating HMM Ensemble Training Algorithms Using Train and Test and Condition Number Criteria.” To Appear in the Journal of Pattern Analysis and Applications, 2003.

[9] D. J. C. Mackay, *Ensemble Learning for Hidden Markov Models, Technical report*, Cavendish Laboratory, University of Cambridge, UK, 1997.

[10] A. Stolcke and S. Omohundro. *Hidden Markov Model induction by Bayesian model merging*, Advances in Neural Information Processing Systems, pp. 11-18, Morgan Kaufmann, San Mateo, United States of America, CA1993.

[11] Nianjun Liu, Brian C. Lovell and Peter J. Kootsookos, ”Evaluation of HMM training algorithms for Letter Hand Gesture Recognition” Proceedings of IEEE International Symposium on Signal Processing and Information Technology. December 14-17, 2003, Darmstadt, Germany. Accepted, In Press.