Understanding HMM Training for Video Gesture Recognition

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

Abstract

When developing a video gesture recognition system to recognise letters of the alphabet based on hidden Markov Model (HMM) pattern recognition, we observed that by carefully selecting the model structure we could obtain greatly improved recognition performance. This led us to the questions: Why do some HMMs work so well for pattern recognition? Which factors affect the HMM training process? In an attempt to answer these fundamental questions of learning, we used simple triangle and square video gestures where good HMM structure can be deduced analytically from knowledge of the physical process. We then compared these analytic models to models estimated from Baum-Welch training on the video gestures. This paper shows that with appropriate constraints on model structure, Baum-Welch reestimation leads to good HMMs which are very similar to those obtained analytically. These results corroborate earlier work where we show that the LR banded HMM structure is remarkably effective in recognising video gestures when compared to fully-connected (ergodic) or LR HMM structures.

1. Introduction

Hidden Markov Models (HMMs) have been used prominently and successfully in speech recognition [5] and hand writing recognition [3] during in the last decades. Given these successes, HMMs have been widely adopted for computer vision and pattern recognition. Starner [7] used Hidden Markov Models for visual recognition of America Sign Language(ASL). Lee and Kim [2] designed an HMM-based threshold model approach for recognition of ten gestures to control PowerPoint slides.

Many researchers apply Baum-Welch and other exemplar-driven algorithms for training HMM models, but use fairly arbitrary selections of fundamental parameters such as structure and order. Finding reliable criteria for the selection of the best HMM parameters is largely an open question — yet these parameters have a severe impact on recognition rates. Due to the lack of guidance for choosing model structure and other parameters, researchers resort to performance evaluations over wide ranges of parameters in the hope of finding good solutions.

When developing a system to recognise video gestures for the 26 letters of the alphabet using HMMs [4], we found that by changing the model structure from Fully-Connected (ergodic) to Left-Right and then finally to Left-Right Banded (LRB), we achieved marked improvement in average recognition rates. Indeed, LRB yielded 97.3% correct recognition on unseen gestures from a database of 780 video gestures — the error rate was 3 times lower than obtained with conventional FC and LR models. These improvements were so startling that we decided to investigate the fundamental problem of model selection in this paper. The questions we are starting to address in this research are: Why do some HMMs work well and others not? Which factors affect the HMM training process? How best to design the model structure and parameters? Some recent work has made progress in this direction [1, 6], but more is required.

In this paper, we present an analytic calculation method for a class of HMMs. This is of particular interest because although no training is required, quite good models are obtained. We show that the analytic models are a good place to start for finding the globally optimal model and can be applied in a novel way for improved HMM training.

2. Direct Computation Method

In this paper we focus on the LRB model structure because of its good performance in video gesture recognition. LRB is a simple HMM structure (Figure 2) that can be described by a single linear chain with only self and next state transitions. With this structure, we can segment gestures (i.e., segment the observation sequence) according to state duration time. A simple way is to evenly segment by the number of states, and then estimate the A matrix from the

Figure 1. Hand gestures for Triangle and Square

duration equation. Because the actual letter gestures varied so much from one to another, we used two much simpler gestures for the purposes of this study: Triangle and Square (Figure 1). The reason for this is that it is easy to partition a Triangle into 3 equal duration parts (states) or a Square into 4 equal duration parts.

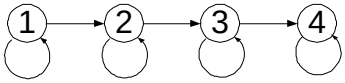


Figure 2. Left-right Banded Structure

In both cases, the state characterizes the expected angle of hand movements to draw each side of a perfect triangle or square. The measurements are the positions of the hand in the video image which are subject to measurement noise and correspond to the HMM state path. From these measurements, the observation sequences are the quantized angles of each hand movement from frame to frame. The B matrix can be viewed as the noise distribution around the state's mean angle value.

2.1. A Matrix Computation

The A matrix is computed using the state duration equation 1. The observation sequence (T) is segmented evenly and the duration time is same for each state. For example, if 3 states are used, the observation sequence length (T) is 24, and the duration (d) is 24/3=8, then $a_{ii} = 0.875$, and because the row sum is 1, then the other value is 0.125. As there is no next state for state 3, $a_{33} = 1$;

$$\bar{d}_i = \frac{1}{1 - a_{ii}} \tag{1}$$

2.2. B Matrix Computation

Each segment corresponds to one state, which ideally corresponds to one observation symbol. However in a real system, noise creates a dispersed distribution of observations from the state. So the B matrix can be directly calculated using one of the following probability distribution estimation methods.

2.2.1 Histogram estimation

Since there are 18 observation symbols and the training set has 20 observation sequences $(T \times 20)$, we segment the training set by the number of states N. Each segment is $(T/N \times 20)$ which is related to its state. We determine the histograms and use them as the elements of the observation probability matrix B.

2.2.2 Fitted Gaussian Distribution

Random variables with unknown distributions are often assumed to be Gaussian. After computing the mean μ and standard deviation σ of the data set, we can easily fit a Gaussian distribution. Since the training data sets are not large, noise values may have a great effect on the sample variance and mean. In this case, the maximum value may be more reliable than the mean and variance can be determined by ensuring the pdf integration to 1 (Max-Value Gaussian).

2.2.3 Von Mises Distribution

As we are dealing with angular data (observation angle), it is more appropriate to use circular distributions. The von Mises distribution is a circular analog of the normal distribution on a line with a mean direction a and concentration parameter b. For small b, it tends to a uniform distribution and for large b it tends to a Normal Distribution with variance $1/b$. Its continuous distribution defined on the range $x \in [0, 2\pi]$ with probability density function:

$$P(x) = \frac{e^{b\cos(x-a)}}{2\pi I_0(b)} \tag{2}$$

where $I_0(x)$ is a modified Bessel function of the first kind of order 0. Here, $a \in [0, 2\pi]$ is the mean direction and $b > 0$ is a concentration parameter.

3. Half and Full Baum Welch Training

We used the traditional Baum Welch (BW) [5] algorithms to train the HMM models. The model structure applied here is always Left-Right Banded (LRB). For the training process, we used two methods. One is Half training, which is to keep the A matrix unchanged, and only train the B matrix. In this way, we can estimate how much errors in the value of the A matrix can affect the final trained output. Full training is the traditional method, which trains both the A and B matrix together.

4. Experiments on Two Simple Gestures

The two gestures we use are actually the trajectories of the hand movements forming a triangle and a square. The

implementation of the hand trajectory analysis were presented in previous work [4]. Along each trajectory, the orientation of each of the 25 hand movements is computed and quantized to one of 18 discrete symbols to form the discrete observation sequences. Then the discrete observation sequence $(T = 24)$ is used as input to a Hidden Markov Model classification module for training and testing. There were 20 training samples and 10 test samples for each of the two gestures; so there 60 gesture videos in the database in total.

4.1. A Matrix Comparison

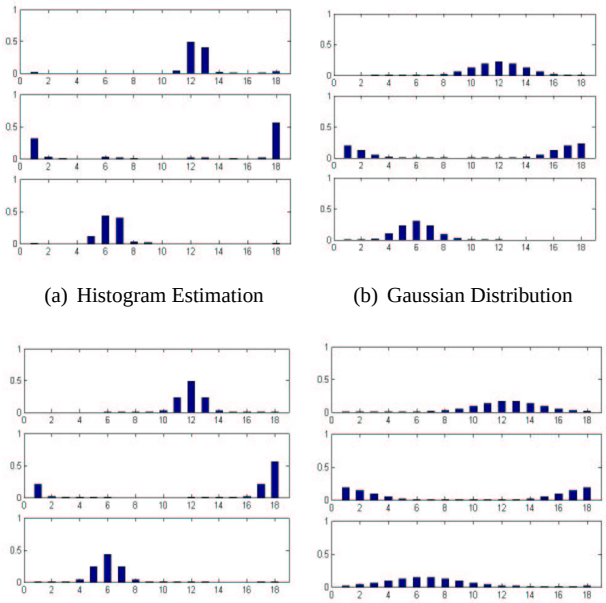
Figure 3 gives the list of A matrices. The left column is for the Triangle, and the right column is for the Square. The first row is computed analytically based on equal duration time for each leg of the trajectory by (1). The second row gives the output from Baum-Welch training on the video data using random LRB intial models (IMs). In Half training, the analytic A matrix (standard A matrix) is used as the initial model and kept unchanged during the training process. For full training with a pre-computed IM, the analytic method is used to initialise the A matrices. For full training with random IM, the initial A matrices are also generated randomly. We trained the HMMs 20 times and averaged the results. The models resulting from Full training with the pre-computed IM all stay very close to the standard A matrix (implying a local maximum has been reached), while the models from Full training using a random IM yield 11 results which are quite close to the standard A matrix, whilst the other 9 were very different.

A Matrix	Triangle (3 states)	Square (4 States)
Directly Compute	0.875 0.125 0 0 0.875 0.125 0 0 1	0.83 0.17 0 0 0 0.83 0.17 0 0 0 0.83 0.17 0 0 0 1
BW-training From Random IM	0.87 0.13 0 0 0.87 0.13 0 0 1	0.85 0.15 0 0 0 0.83 0.17 0 0 0 0.85 0.15 0 0 0 1

Figure 3. Comparing A Matrix Between the two Methods

4.2 B Matrix Comparison

In section 4.2, we show the experimental results for computing the B matrix by various methods, including different distribution fitting algorithms and different training methods with multiple topologies on the training data set.



(a) Histogram Estimation (b) Gaussian Distribution

(c) Max-Value Gaussian (d) von Mises Distribution

Figure 4. B Matrix estimation by multiple methods (Triangle)

4.2.1 Pre-computed B Matrix Comparison

Figure 4 shows B Matrices computed from the distribution estimation methods. The histogram is not very accurate due to lack of samples. A fitted gaussian distribution is a way of handling this problem, but because the data set is not large enough, it did not represent the observation probability B Matrix well. The Max-Gaussian method matches the histogram better, and the distribution tails are more suppressed.

4.2.2 B Matrix Training Comparison

We show the B matrix trained outputs for the Triangle gesture in figure 6. We use four ways to train the B Matrix: Half, Full training and by random and pre-computed Initial Models (IM). The aim of the experiment was to check the degree of match of B between training outputs and the pre-computed IM. The B matrix directly computed from the histogram is defined as the standard B matrix. We trained 20 times on each of the four methods. The quality of match is shown in Figure 5.

4.2.3 Discussion of the Degree of Matching

In order to compare the A and B matrix from different methods, we use the difference parameter *DiffDist* defined by:

$$\textit{DiffDist} = vec(M - M_b)vec(M - M_b)^T. \quad (3)$$

We use the difference of the target matrix(M) and the standard matrix(M_b), and sum the squares of each element in the difference matrix. The result means that the smaller the *DiffDist* value is, the closer the match to the standard matrix is. Figure 5 shows the matching degree of A and B. After analyzing this, we make the following observations.

Method	A matrix *DiffDist* (Triangle)	B matrix *DiffDist* (Triangle)	A Matrix *DiffDist* (Square)	B Matrix *DiffDist* (Square)
Half-BW-Ran	0	0.4654	0	0.7284
Half-BW-Pre	0	0.0066	0	0.0698
BW-Ran	0.0708	0.8632	0.1314	1.4370
BW-Pre	2.1315e-005	0.0059	8.1856e-004	0.0676

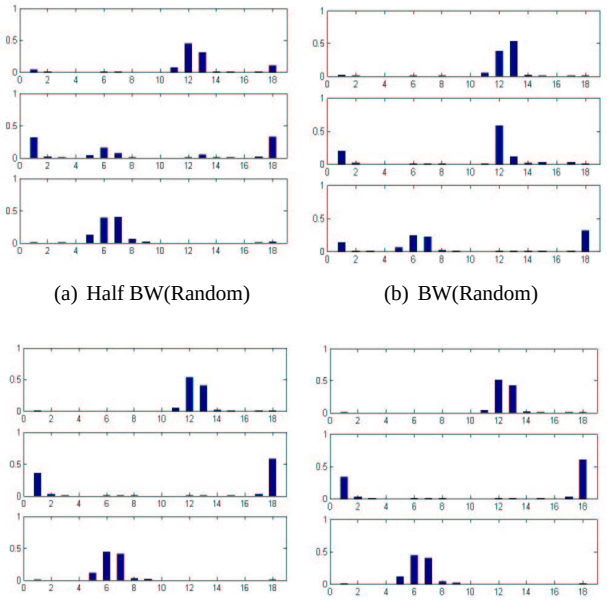
Figure 5. A and B Matrix matching degree by *DiffDist*

1. When training A and B together by Baum Welch with the pre-computed initial parameters, the trained models match both of the standard matrices very well for both triangle and square. However the result using the random IM does not match. The degree of match for the A matrix on the pre-computed IM is 2.1315e-005 and 8.1856e-004 for triangle and square respectively, and for the B Matrix it is 0.0059 and 0.0676. While the degree of matching on the random IM is 0.0708 (Triangle) and 0.1314 (Square) for A matrix, and 0.8632 and 1.4370 for B matrix. It was interesting to note that the IM from the histogram seems to be a good place to start to reach the global optimum.

2. For Half training, the output B matrix on the pre-computed IM matches the standard B matrix much more than on the random IM. In the triangle gesture, the Half-BW matching degree with the pre-computed IM is 0.0066, and the one using the random method is 0.4654. It further justifies the choice of initial models from the histogram distribution.

5. Conclusion

The paper uses two simple gestures to investigate the Baum-Welch training performance. The A Matrix is computed based on the Left-Right Banded HMM model and equal duration in states. The B Matrix is treated as a probability distribution and solved through the corresponding equations on the real data. The A and B matrices obtained from the direct computation are quite well matched with the ones trained from the Baum-Welch training methods with the pre-computed initial models. However the training output from the randomly initialised models does not match well in some cases. We have proposed some novel ways to estimate the initial model from direct computation.



(a) Half BW(Random) (b) BW(Random)

(c) Half BW(pre-computed) (d) BW(pre-computed)

Figure 6. B Matrix training results

References

[1] R. Davis and B. C. Lovell. Comparing and evaluating hmm ensemble training algorithms using train and test and condition number criteria. *Journal of Pattern Analysis and Applications*, 2003.

[2] 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, Oct 1999.

[3] J. Lee, J.Kim, and J. Kim. Data-driven design of hmm topology for online handwriting recognition. *International Journal of Pattern Recognition and Artifical Intelligence*, 15(1):107–121, 2001.

[4] N. Liu, B. Lovell, and P. Kootsookos. Evaluation of hmm training algorithms for letter hand gesture recognition. *IEEE International Symposium on Signal Processing and Information Technology*, December 2003.

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

[6] T.Caelli and B.McCane. Components analysis of hidden markov models in computer vision. *12th International Conference on Image Analysis and Processing*, September 2003.

[7] T.Starner and A.Pentland. Real-time american sign language recognition. *IEEE Trans, On Pattern Analysis and Machine Intelligence*, 20:1371–1375, Dec 1998.