MAXIMUM LIKELIHOOD TRAINING OF THE EMBEDDED HMM FOR FACE DETECTION AND RECOGNITION

Ara V. Nefian and Monson H. Hayes III

Center for Signal and Image Processing
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
{ara, mhh3}@eedsp.gatech.edu

ABSTRACT

The embedded hidden Markov model (HMM) is a statistical model that can be used in many pattern recognition and computer vision applications. This model inherits the partial size invariance of the standard HMM, and, due to its pseudo two-dimensional structure, is able to model two-dimensional data such as images, better than the standard HMM. In this paper we describe the maximum likelihood training for the continuous mixture embedded HMM and present the performance of this model for face detection and recognition. The experimental results are compared with other approaches to face detection and recognition.

1. INTRODUCTION

This embedded HMM or pseudo two dimensional HMM, first introduced for character recognition by Kuo and Agazzi [1], has a large potential for many pattern recognition applications that involve two dimensional data. One very important application where the embedded HMM can be used is face modeling for detection and recognition. Face detection and recognition systems have many applications varying from identification systems (control the access of people into restricted areas) to multimedia applications (face recognition from video or from a photography album). Previous attempts to use HMM for face modeling include the left-to-right HMM [2], [3] and the HMM with end-of-line states described in [4]. In this paper we describe the maximum likelihood training of the embedded HMM and present a method for face detection and recognition using this model.

2. THE EMBEDDED HMM

An embedded HMM is a generalization of a HMM where each state in a one-dimensional HMM is itself an HMM. Thus, an embedded HMM consists of a set of *super states* along with a set of *embedded* states. The super states model the two-dimensional data along one direction, while the embedded HMMs model the data along the other direction. Specifically, the elements of an embedded HMM are:

1. A set of N_0 super states.
2. The initial super state probability distribution, $\mathbf{\Pi}_0 = \{\pi_{0,i}\}$, where $\pi_{0,i}$ is the probability of being in super state i at time zero.
3. The state transition matrix between the super states, $\mathbf{A}_0 = \{a_{0,ij}\}$, where $a_{0,ij}$ is the probability of making a transition from super state i to super state j.
4. In an embedded HMM, each super state k is itself a standard HMM defined by the parameter set $\Lambda^k = (\mathbf{\Pi}_1^k, \mathbf{A}_1^k, \mathbf{B}^k)$, where $\mathbf{\Pi}_1^k$ is the initial state probability distribution of the embedded states, $\mathbf{A}_1^k$ is the state transition matrix for the embedded states, and $\mathbf{B}^k$ is the probability distribution matrix of the observations. With a *continuous mixture* embedded HMM, the observations are characterized by a continuous probability density function, which are taken to be finite Gaussian mixtures of the form,

$$b_i^k(\mathbf{O}_{t_0,t_1}) = \sum_{m=1}^{M_i^k} c_{im}^k N(\mathbf{O}_{t_0,t_1}, \mu_{im}^k, \mathbf{U}_{im}^k)$$

where c_{im}^k is the mixture coefficient for the mth mixture in state i of super state k, and $N(\mathbf{O}_{t_0,t_1}, \mu_{im}^k, \mathbf{U}_{im}^k)$ is a Gaussian density with a mean vector μ_{im}^k and covariance matrix $\mathbf{U}_{im}^k$. Note that for the observation vector $\mathbf{O}_{t_0,t_1}$ we have two subscripts, t_0 and t_1. We denote the sequence of observation vectors as $\mathbf{O}_{t_0} = \{\mathbf{O}_{t_0,t_1} |\ 0 \le t_1 < T_1\}$. and the two-dimensional sequence of observation vectors as $\mathbf{O} = \{\mathbf{O}_{t_0} |\ 0 \le t_0 < T_0\}$.

Using a shorthand notation, an embedded HMM is defined by the triplet $\lambda = (\mathbf{\Pi}_0, \mathbf{A}_0, \Lambda^k)$.

3. MAXIMUM LIKELIHOOD TRAINING OF THE EMBEDDED HMM

In this section we describe the re-estimation equations for the maximum likelihood training algorithms of the embedded HMM. Because the evaluation algorithms play an important role in the understanding of the re-estimation equation we will also discuss them briefly.

3.1. The evaluation algorithms

An efficient algorithm for the computation of $P(\mathbf{O}|\lambda)$ is obtained if the forward variable for the sequence $\mathbf{O}_0, \mathbf{O}_1, \ldots, \mathbf{O}_{T_0}$

is defined as:

$$\alpha_{t_0}(i) = P(\mathbf{O}_1, \mathbf{O}_1, \ldots, \mathbf{O}_{t_0}, q^0_{t_0} = i|\lambda)$$

where $q^0_{t_0}$ is the super state corresponding to $\mathbf{O}_{t_0}$. The forward variable $\alpha_{t_0}(i)$ describes the probability of the partial sequence $\mathbf{O}_0, \mathbf{O}_1, \ldots, \mathbf{O}_{t_0}$ and super state i given the model λ. Therefore, $P(\mathbf{O}|\lambda)$ can be computed as:

$$P(\mathbf{O}|\lambda) = \sum_{all\ i} \alpha_{T_0}(i)$$

The forward variable $\alpha_{t_0}(i)$ is computed iteratively from its previous values and the probability of $\mathbf{O}_{t_0}$ given the super state i, $P(\mathbf{O}_{t_0}|q^0_{t_0} = i, \lambda)$:

$$\alpha_{t_0+1}(i) = [\sum_i \alpha_{t_0}(i) a_{0,ij}] P(\mathbf{O}_{t_0}|q^0_{t_0} = i, \lambda)$$

It is important to notice that, as with the embedded Viterbi algorithm [1], the delay introduced by the computation of the forward algorithm can be significantly reduced if a parallel architecture is used. In such a parallel implementation, all $P(\mathbf{O}_{t_0}|q^0_{t_0} = i, \lambda)$ can be computed at the same time using the forward backward algorithms for the standard HMM [5]. Similarly a backward variable can be defined and used to compute $P(\mathbf{O}|\lambda)$.

3.2. The re-estimation algorithm

Let $\underline{\mathbf{O}} = \{\mathbf{O}^1, \ldots, \mathbf{O}^r, \ldots, \mathbf{O}^R\}$, be a set of R independent two-dimensional observation vectors. The objective of the training algorithm is to iteratively search for the set of parameters of the embedded HMM that maximize $P(\underline{\mathbf{O}}|\lambda) = \prod_r P(\mathbf{O}^r|\lambda)$, which is equivalent to maximizing the auxiliary function:

$$Q(\lambda, \lambda') = \frac{1}{P(\underline{\mathbf{O}}|\lambda)} \sum_{\mathbf{q}} P(\underline{\mathbf{O}}, \mathbf{q}|\lambda) \log P(\underline{\mathbf{O}}, \mathbf{q}|\lambda') \qquad (1)$$

with respect to λ. The re-estimated parameters of the embedded HMM are derived using a variant of the EM algorithm to minimize the auxiliary function in Equation 1. The re-estimated parameters are given by the following equations:

$$\pi_{0,i} = \frac{\sum_r \gamma_0^{'(r)}(i)}{\sum_r \sum_i \gamma_0^{'(r)}(i)}$$

$$a_{0,ij} = \frac{\sum_r \sum_{t_0} \gamma_{t_0}^{'(r)}(i,j)}{\sum_r \sum_{t_0} \gamma_{t_0}^{'(r)}(i)}$$

$$\pi^i_{1,j} = \frac{\sum_r \sum_{t_0} \gamma^{(r)}_{t_0,0}(i,j)}{\sum_r \sum_{t_0} \gamma_{t_0}^{'(r)}(i)}$$

$$a^i_{1,jl} = \frac{\sum_r \gamma^{(r)}_{t_0,t_1}(i,j,l)}{\sum_r \sum_{t_0} \sum_{t_1} \gamma^{(r)}_{t_0,t_1}(i,j)}$$

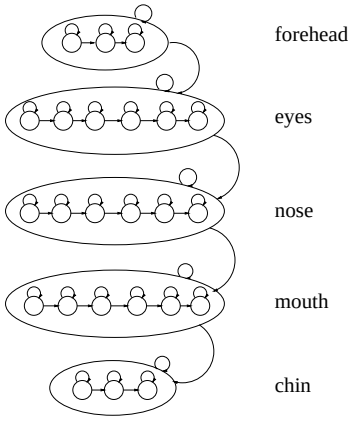

Figure 1: Embedded HMM for face

$$c^i_{jm} = \frac{\sum_r \sum_{t_0} \sum_{t_1} \zeta^{(r)}_{t_0,t_1}(i,j,m)}{\sum_r \sum_{t_0} \sum_{t_1} \gamma^{(r)}_{t_0,t_1}(i,j)}$$

$$\mu^i_{jm} = \frac{\sum_r \sum_{t_0} \sum_{t_1} \zeta^{(r)}_{t_0,t_1}(i,j,m) \mathbf{O}^r_{t_0,t_1}}{\sum_r \sum_{t_0} \sum_{t_1} \zeta^{(r)}_{t_0,t_1}(i,j,m)}$$

$$\mathbf{U}^i_{jm} = \frac{\sum_r \sum_{t_0} \sum_{t_1} \zeta^{(r)}_{t_0,t_1}(i,j,m)(\mathbf{O}^r_{t_0,t_1} - \mu^i_{jm})(\mathbf{O}^r_{t_0,t_1} - \mu^i_{jm})^T}{\sum_r \sum_{t_0} \sum_{t_1} \zeta^{(r)}_{t_0,t_1}(i,j,m)}$$

where,

$$\gamma_{t_0}^{'(r)}(i) = P(q^0_{t_0} = i|\mathbf{O}^r, \lambda)$$

$$\gamma_{t_0}^{'(r)}(i,j) = P(q^0_{t_0-1} = i, q^0_{t_0} = j|\mathbf{O}^r, \lambda)$$

$$\gamma^{(r)}_{t_0,t_1}(i,j) = P(q^0_{t_0} = i, q^1_{t_0,t_1} = j|\mathbf{O}^r, \lambda)$$

$$\gamma^{(r)}_{t_0,t_1}(i,j,l) = P(q^0_{t_0} = i, q^1_{t_0,t_1-1} = j, q^1_{t_0,t_1} = l|\mathbf{O}^r, \lambda)$$

$$\zeta^{(r)}_{t_0,t_1}(i,j,m) = P(q^0_{t_0,t_1} = i, q^1_{t_0,t_1} = j, k_{t_0,t_1} = m|\mathbf{O}^r, \lambda)$$

In the above equations, $q^0_{t_0,t_1}$, $q^1_{t_0,t_1}$ and k_{t_0,t_1} represent the super state, the embedded state and the mixture corresponding to the observation vector $\mathbf{O}_{t_0,t_1}$. The above variables are obtained from the forward and backward variables as described in more detail in [6].

4. THE FACE MODEL

The structure of the embedded HMM used for both face detection and recognition is illustrated in Figure 1. The observation sequence for a face image is formed from image blocks that are extracted by scanning the image from left-to-right and top-to-bottom. Adjacent image blocks overlap in the vertical direction, and in the horizontal direction.

	Experiment 1		Experiment 2	
	DR	FA	DR	FA
Embedded HMM				
2D - DCT	91.7%	$\frac{16}{426,187,008}$	91.2%	$\frac{43}{1,278,561,024}$
KLT	96.3%	$\frac{7}{426,187,008}$	91.5%	$\frac{56}{1,278,561,024}$
HMM				
2D-DCT	79.2%	$\frac{10}{84,188,160}$	68.3%	$\frac{39}{252,564,480}$
KLT	81.3%	$\frac{9}{84,188,160}$	72.6%	$\frac{45}{252,564,480}$

Table 1: Comparison of the detection rate (DR) and false alarms (FA) in different experiments obtained using the standard HMM and the embedded HMM

The observation vectors consist of either four KLT coefficients (corresponding to the largest eigenvalues) or six 2D-DCT coefficients (corresponding to a 2×3 window around the lowest frequencies in the 2D DCT domain) that correspond to each 8×10 pixel block (6×8 overlap). The blocks are extracted from the training images by scanning them from left to right and top to bottom.

5. FACE DETECTION

To detect a face in a test image, first a face model is trained using the doubly embedded Viterbi segmentation followed by the maximum likelihood training procedure described above. The initial set of parameters are obtained from uniformly segmenting the face images according to the structure of states and super states of the embedded HMM. The training images consist of 400 frontal faces from the ORL database [4]. The embedded HMM face model obtained after training is used to compute the doubly embedded Viterbi score for each rectangular pattern in a test image. The rectangular patterns for which the likelihood score exceeds a fixed threshold are taken as faces. To reduce the number of false alarms, the face candidates that overlap with rectangular patterns of higher score are discarded. The above face detection algorithm has been tested in two experiments using both KLT and 2D-DCT based features. In the first experiment the test images consists of 144 images in the MIT database showing frontal faces with variations in illumination and size (by a factor of two). In the second experiment the test images consist in 432 images from the same database where images showing faces with variations in pose (rotations in the image plane) where added. Table 1 presents the detection results in both experiments using the embedded HMM and the standard HMM. The false alarms are reported along with the total number of patterns extracted from the test images. It is important to notice that for both the embedded HMM and the standard HMM the KLT features performed better than the 2D-DCT. This is due to the optimal decorrelation properties of the KLT, and to the fact that the KLT basis was obtained from face images. From the above experiments, it is clear that the embedded HMM outperforms the standard HMM for face detection.

	Approach	Recognition Rate
1	Auto Association and Classification NN [7]	20%
2	Dynamic Link Matching [8]	80%
3	Eigenface [9]	80%
4	HMM [2]	85%
5	VFR model [10]	92.5%
6	HMM end-of-line-states [4]	90-95%
7	PDBNN [11]	96%
8	Convolutional NN [12]	96.2%
9	Embedded HMM	100%

Table 2: Comparison of the face recognition rate for different approaches tested on the ORL database

6. FACE RECOGNITION

The face recognition approach described in this paper was tested on images from the ORL database and a new database from Georgia Tech. The observation vectors are obtained in the same manner as discussed for detection, using either 2D-DCT or KLT based coefficients. Different instances of the same subject in the database are used to train the embedded HMM corresponding to one face. To recognize a face, the doubly embedded Viterbi score is computed for the test image given the models corresponding to the faces in the database, and the highest score is selected to reveal the identity of the test image.

Table 2 compares some of the face recognition approaches described in the literature and tested on the same database. As shown in this table, the embedded HMM produces the best face recognition results on the ORL database. The perfect recognition rate is obtained when each state of the embedded HMM was modeled by a mixture of three Gaussian density functions. We have also tested our approach on a new database built at Georgia Tech. The database contains 450 images of 50 people (15 images per person) both males and females from different ages and races. Most of the images were taken in two or three sessions over a period of three months such that changes in illumination and facial applearnce become more evident. Each image in the database shows one face in a complex background. The faces have strong variations in size and orientation (rotations in the image plane and the plane perpendicular to the image plane) and facial expressions. We have tested our system using 10 images in the training set for each person and use the remaining five face images for testing. The images in both the testing and training set were manually croped. The recognition results, using six DCT-based observation vectors and three mixtures per state show 87% correct recognition and outperform the eigenface method (40 eigenfaces) by almost 20%. (correct recognition 68%). Figure 2 shows some of the recognition results on the Georgia Tech database. The crossed images represent misclassifications while the rest of the images represent correct recognition. Table 3 compares different HMM-based methods for face recognition in terms of recognition rate (tested on the ORL database) and complexity (in terms of additions).

Figure 2: Face recognition results tested on the Georgia Tech database

	Recog. Rate	Complexity (additions)
HMM [2]	85%	$N_0^2 T_0$
HMM [4]	90-95%	$(\sum_{k=1}^{N_0} N_1^{(k)})^2 T_0 T_1$
Emb. HMM	98%	$\sum_{k=1}^{N_0} (N_1^{(k)})^2 T_1 T_0 + {N_0}^2 T_0$

Table 3: Comparison of the face recognition rate and numerical complexity for different HMM-based approaches

To make a fair comparison with Samaria's model, which uses a Gaussian density for state modeling, the recognition rates in Table 3, were obtained by removing the mixture components. In this case the recognition rate of the embedded HMM drops from 100% to 98%, but still achieves the best performance among the HMM-based approaches. Although more complex than the standard HMM, the embedded HMM is less complex than the HMM with end of line states. Furthermore, due to the parallel structure of the embedded Viterbi algorithm used for recognition, the time delay introduced by the embedded HMM can be significantly decreased and approaches the delay found in the standard HMM.

7. CONCLUSIONS

This paper describes the maximum likelihood training for a continuous mixture embedded HMM, and demonstrates the efficiency of this model for face detection and recognition. The embedded HMM, due to its pseudo two-dimensional structure, outperforms the standard HMM and the HMM with end of line states for face detection and recognition where data is two dimensional.

Unlike the HMM with end of line states, the embedded HMM provides a natural set of initial parameters for the training, and fast training and recognition algorithms that can also be implemented in a parallel architecture.

Compared with the template-based methods for face detection and recognition, the embedded HMM is more flexible with respect to variations in scale, natural deformations and allows for a faster implementation of the face detection algorithm due to the breaking of the face templates in image blocks which are processed to obtain the observation vectors.

8. REFERENCES

[1] S. Kuo and O. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-D Hidden Markov Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 842–848, August 1994.

[2] F. Samaria and S. Young, "HMM based arhitecture for face identification," *Image and Vision Computing*, vol. 12, pp. 537–543, October 1994.

[3] A. V. Nefian and M. H. Hayes, "Face detection and recognition using Hidden Markov Models," in *International Conference on Image Processing*, vol. 1, pp. 141–145, October 1998.

[4] F. Samaria, *Face Recognition Using Hidden Markov Models*. PhD thesis, University of Cambridge, 1994.

[5] L. Rabiner and B. Huang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[6] A. Nefian, *A hidden Markov model based approach for face detection and recognition*. PhD thesis, Georgia Tech, 1999.

[7] G. Cotrell and M. Fleming, "Face recognition using unsupervised feature extraction," in *Proceedings International neural Network Conference*, pp. 322–325, 1990.

[8] M. Lades, J.Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, and R. Wurtz, "Distorsion invariant object recognition in the dynamic link architecture," *IEEE Transcations on Computers*, vol. 42, no. 3, pp. 300–311, 1993.

[9] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proceedings of International Conference on Pattern Recognition*, pp. 586 – 591, 1991.

[10] J. Ben-Arie and D. Nandy, "A volumetric/iconic frequency domain representation for objects with application for pose invariant face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 449–457, May 1998.

[11] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Network*, vol. 8, pp. 114–132, January 1997.

[12] A. Lawrence, C. Giles, A. Tsoi, and A. Back, "Face recognition : A convolutional neural network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.