EMBEDDED BAYESIAN NETWORKS FOR FACE RECOGNITION

Ara V. Nefian

Intel Corporation
Microprocessor Research Labs
Santa Clara , CA 95052
ara.nefian@intel.com

ABSTRACT

The embedded Bayesian networks (EBN) introduced in this paper, are a generalization of the embedded hidden Markov models previously used for face and character recognition. An EBN is defined recursively as a hierarchical structure where the ”parent” node is a Bayesian network (BN) that conditions the EBNs or the observation sequence that describes the nodes of the ”child” layer. With an EBN, one can model complex N-dimensional data, avoiding the complexity of N-dimensional BN while still preserving their flexibility and partial scale invariance. In this paper we present an application of the EBNs for face recognition and show the improvement of this approach versus the ”eigenface” and the embedded HMM approaches.

1. INTRODUCTION

The work presented in this paper is motivated by the need for practical statistical models with N-dimensional dependencies, in particular with two-dimensional dependencies used for image analysis. While the hidden Markov models (HMM) are very successful in speech recognition or gesture recognition where data dependency is one-dimensional over time, an equivalent N-dimensional HMM has been shown to be impractical due to its complexity that grows exponentially with the size of the data [1]. For image recognition, and in particular face recognition [2] were data is essentially two-dimensional, template based approaches using principal component analysis ([3], [4]), linear discriminant analysis ([5]), neural networks ([6], [7]), and matching pursuit [8] showed improved results over the early geometric feature representations ([9]). However, these approaches cannot generalize over a wide variation in scale, orientation, or facial expression. In recent years, several approaches to approximate a 2D HMM with computationally practical models have been investigated such as the pseudo 2D HMM or the embedded HMM used in character recognition [1] or face recognition [10], [11]. These models significantly reduce the error rate of the earlier HMM-based face recognition approaches [11]. In [12], Jia and Gray developed an efficient approximation for the training and recognition of the 2D HMM applied to text image analysis. In this paper we introduce a family of embedded Bayesian networks (EBN) and investigate their performance for face recognition. The EBN generalize the embedded HMM by allowing each HMM to be replaced by any arbitrary Bayesian Network (BN). Specifically, we introduce a family of EBN that builds on existing dynamic BN such as the HMM or the coupled HMMs ([13]) and compare their face recognition performance with some of the existing approaches.

2. THE COUPLED HIDDEN MARKOV MODEL

The coupled HMM (CHMM), can be seen as a collection of HMMs, one for each data stream, where the discrete nodes at time t for each HMM are conditioned by the discrete nodes at time $t-1$ of all the related HMMs. Figure 1 illustrates a CHMM where the squares represent the hidden discrete nodes while the circles describe the continuous observable nodes. Let C be the number of channels of a CHMM, and $\mathbf{i} = [i_1, \ldots, i_C]$ be the state vector describing the state of the hidden nodes in channels $1, \ldots, C$ $\mathbf{q}_t = [q_t^1, \ldots q_t^C]$ at one particular time instance t. The elements of the coupled HMM are $\pi_0^c(i_c) = P(\mathbf{O}_0^c|q_t^c = i_c)$ the initial state probability of state i_c in channel c, $a^c_{i_c|\mathbf{j}} = P(q_t^c = i_c|q_{t-1} = \mathbf{j})$, the state transition probability from state $\mathbf{j}$ to state i_c in channel c, and $b_t^c(i_c) = P(\mathbf{O}_t^c|q_t^c = i_c)$ the observation likelihood give the state i_c in channel c.

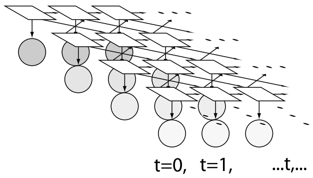

Figure 1: A coupled hidden Markov model for image recognition.

3. A FAMILY OF EMBEDDED BAYESIAN NETWORKS

The embedded Bayesian network (EBN) is a hierarchical statistical model consisting of several layers, each, with the exception of the lowest layer, being defined by a set of EBN. The lowest layer consists of a set of observation vectors. The parameters of each EBN within the same layer are independent from each other, while their parameters depend on their ”parent” EBN in the upper layer. The EBN introduced in this paper for face recognition consists of two layers, one layer for each data dimension. Each layer is described by either a set of HMMs or a CHMMs, resulting in a family of four EBNs shown in Figure 2: the embedded HMM (EHMM) where both layers are described by HMMs, the HMM-CHMM where the upper layer is an HMM and the lower layer consist of a set of CHMMs , the CHMM-HMM where the upper layer is a CHMM and the lower layer consists of a set of HMMs, and the embedded CHMM (ECHMM) where both layers are described by CHMMs. Since the HMM can be seen as a CHMM with one data stream, the ECHMM is a generalization of all the models in its family. For the purpose of simplicity this paper will only give the formal definition and describe the training algorithm for the most general model of the family, namely the ECHMM. The formal definition of the parameters of a two-layer ECHMM, given below, can be extended to any number of layers. Throughout this paper we will refer to the channels, nodes and states of the ”parent” CHMM as the *super channels*, the *super nodes*, and the *super states*of the ECHMM. For simplicity assume that all CHMMs in layer l have the same number of channels C_l. The elements of a two layer ECHMM are:

- the initial super state probability in super channel $s = 1, \dots C_0$, $\pi^{s}_{0,0}$
- the super state transition probability from the sequence of states $\mathbf{j} = [j_1, \dots j_{C_0}]$ to state i_s in super channel s, $a^{s}_{0,i_s|\mathbf{j}}$.
- for each super state k in the super channel s the parameters of the corresponding CHMM are defined as follows:
  - the initial state probability in channel $c = 1, \dots C_1$, $\pi^{s,k,c}_{1,0}$
  - the state transition probability $a^{s,k,c}_{1,i_c|\mathbf{j}}$
  - the observation probabilities $b^{s,k,c}_{t_0,t_1}(j_c)$. In a continuous mixture with Gaussian components, the probability of the observation vector $\mathbf{O}$ is given by:

$$b^{s,k,c}(j_c) = \sum_{m=1}^{M^{s,k,c}_j} w^{s,k,c}_{j,m} N(\mathbf{O}, \mu^{s,k,c}_{j,m}, \mathbf{U}^{s,k,c}_{j,m}) \quad (1)$$

where $\mu^{s,k,c}_{j,m}$ and $\mathbf{U}^{s,k,c}_{j,m}$ are the mean and covariance matrix of the mth mixture of the jth state in the cth channel. $M^{s,k,c}_j$ is the number of mixtures corresponding to the jth state of the cth channel and the weight $w^{s,k,c}_{j,m}$ is the corresponding mixture weight.

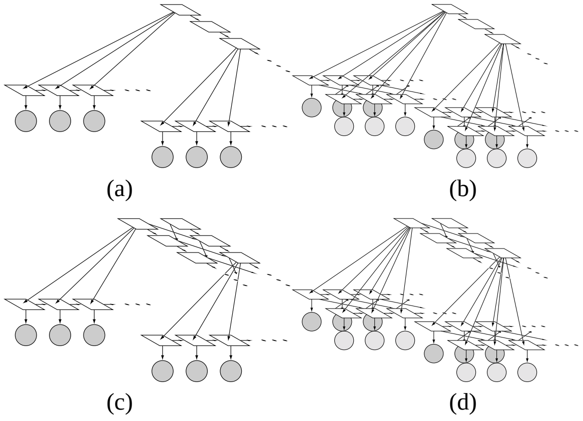
(a)
(b)
(c)
(d)

Figure 2: (a) The embedded HMM , (b) The HMM-CHMM structure , (c) The CHMM-HMM structure, (d)The embedded CHMM

4. THE OBSERVATION VECTORS

The observation sequence for an image is extracted from image blocks of size $L_x \times L_y$ that are obtained by scanning the image from left-to-right and top-to-bottom as illustrated in Figure 3. Adjacent image blocks overlap by P_y rows in the vertical direction, and P_x columns in the horizontal direction. Specifically, with blocks of $L_y = 8$ rows and $L_x = 8$ columns, we used six 2D-DCT coefficients (a 3 × 2 low-frequency array). The use of 2D-DCT coefficients instead of pixel values as observation vectors is justified by the compression and decorrelation properties of the 2D DCT transform for natural images. The resulting array of observation vectors is of size $T_0 \times T_1$, where T_0 and T_1 are the number of observation vectors extracted from the height H and width W of the image.

$$T_0 = \frac{H - L_y}{L_y - P_y} + 1,$$
$$T_1 = \frac{W - L_x}{L_x - P_x} + 1$$

Next consecutive horizontal and vertical observation vectors are grouped together in *observation blocks*. Throughout this paper we will denote $\mathbf{O}_{t_0,s,t_1,c}$ as the t_1th observation vector corresponding to the cth channel within the observation block (t_0, s).

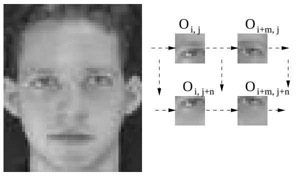

Figure 3: Face image parameterization and blocks extraction

5. THE OPTIMAL STATE SEQUENCE SEGMENTATION

The algorithm described in this section determines the optimal state and super state segmentation of the observation sequence for the ECHMM. This algorithm also referred to as the *Viterbi algorithm for ECHMM* can be applied to all the members of the EBN family discussed in this paper by setting the appropriate number of channels and super channels. For simplicity, we describe here the Viterbi algorithm for a two layer ECHMM.

- for each observation block (t_0, s) we compute the Viterbi algorithm for a HMM [14] or CHMM [15], given the super state i_s of the super channel s. The best super state probability and the optimal state segmentation for the observation block (t_0, s) given the super state i_s of the super channel s is denoted as $P_{t_0,s}(i_s)$ and $\beta_{t_0,s,t_1,c}(i_s)$ respectively.
- Initialization

$$\begin{aligned}\delta_{0,0}(\mathbf{i}) &= \prod_s \pi^{s}_{0,0}(i_s) P_{t_0,s}(i_s) \\ \psi_{0,0}(\mathbf{i}) &= 0\end{aligned}$$

- Recursion

$$\begin{aligned}\delta_{0,t_0}(\mathbf{i}) &= \max_{\mathbf{j}} \{\delta_{0,t_0-1}(\mathbf{j}) \prod_s a^{s}_{0,i_s|j_{s-1},j_s,j_{s+1}} P_{t_0,s}(i_s)\} \\ \psi_{0,t_0}(\mathbf{i}) &= \arg\max_{\mathbf{j}} \{\delta_{0,t_0-1}(\mathbf{j}) \prod_s a^{s}_{0,i_s|j_{s-1},j_s,j_{s+1}} P_{t_0,s}(i_s)\}\end{aligned}$$

- Termination

$$\begin{aligned} P &= \max_{\underline{i}}\{\delta_{T_0}(\mathbf{i})\} \\ \{\alpha_{T_0,1},\ldots,\alpha_{T_0,S}\} &= \arg\max_{\mathbf{i}}\{\delta_{T_0}(\mathbf{i})\} \end{aligned}$$

- Backtracking

$$\begin{aligned} \{\alpha_{t_0,1},\ldots,\alpha_{t_0,S}\} &= \psi_{0,t+1}(\alpha_{t_0+1,1},\ldots,\alpha_{t_0+1,S}) \\ q^0_{t_0,s,t_1,c} &= \alpha_{t_0,s}; \\ q^1_{t_0,s,t_1,c} &= \beta_{t_0,s,t_1,c}(\alpha_{t_0,s}); \end{aligned}$$

In practice, to overcome the underflow problems and the large number of multiplications, all of the above calculations can be used in logarithm form. Table 1 compares the complexity of the Viterbi search for a two-layer EBN with N_0 and N_1 states per node for each of the BN in the upper and lower layer respectively and a CHMM with N_0N_1 states . Together with a significantly smaller complexity the Viterbi algorithm for EBN can be computed in parallel, since the models of the lower level are independent one from the others.

Model	Additions
CHMM	$(N_0N_1)^{(2T_0)}T_1T_0$
EBN	$N_1^{2C_1}N_0T_1T_0 + N_0^{2C_0}T_0\frac{C_0}{C_1}$

Table 1: A comparison of the complexity in terms of additions for the CHMM and EBN

6. TRAINING

To train an EBN, the observation vectors are first extracted from the images in the training set. Throughout this paper we will denote the membership to the rth example image in the training set as super script (r). To train an EBN we proceed as follows:

1. At the first iteration, the array of observation blocks is uniformly segmented into S vertical channels, and the vectors within each super channel are uniformly segmented according to the number of super states in each channel. Next the observation array within each observation block is segmented uniformly according to the number of channels and states of each ”child”. To initialize the mixture components, the observation sequence assigned to each channel c, state j, super state k, and super channel s are assigned to $M_j^{s,k,c}$ clusters using the K-means algorithm.
2. At the next iteration, the uniform segmentation is replaced by the optimal state segmentation algorithm described in section 5. The mixture components in each state j and super state k are determined by assigning the observation $\mathbf{O}^{(r)}_{t_0,s,t_1,c}$, from the rth example in the training set, to the Gaussian component for which the Gaussian density function $N(\mathbf{O}^{(r)}_{t_0,s,t_1,c};\mu^{s,k,c}_{j,m},\mathbf{U}^{s,k,c}_{j,m})$ is highest.
3. The model parameters are then estimated using an extension of the segmental k-means algorithm.

   Specifically, the estimated transition probabilities between super states $\tilde{a}^s_{0,i_s|\mathbf{j}}$ are obtained as follows

   $$\tilde{a}^s_{0,i_s|\mathbf{j}} = \frac{\sum_r\sum_{t_0}\sum_{t_1}\epsilon^{(r)}_{t_0}(s,i_s,\mathbf{j})}{\sum_r\sum_{t_0}\sum_{t_1}\sum_{\mathbf{l}}\epsilon^{(r)}_{t_0}(s,i_s,\mathbf{l})}$$

   where $\epsilon^{(r)}_{t_0}(s,i_s,\mathbf{l})$ is one if a transition from state sequence $\mathbf{l}$ to the super state i_s in super channel s occurs for the observation block (t_0,s) and zero otherwise.

   The estimated transition probabilities between the embedded states $\tilde{a}^{s,k,c}_{1,i_c|\mathbf{j}}$ are obtained as follows,

   $$\tilde{a}^{s,k,c}_{1,i_c|\mathbf{j}} = \frac{\sum_r\sum_{t_0}\sum_{t_1}\theta^{(r)}_{t_0,t_1}(s,k,c,i_c,\mathbf{j})}{\sum_r\sum_{t_0}\sum_{t_1}\sum_{\mathbf{l}}\theta^{(r)}_{t_0,t_1}(s,k,c,i_c,\mathbf{l})}$$

   where $\theta^{(r)}_{t_0,t_1}(s,k,c,i_c,\mathbf{l})$ is one if in the observation block (t_0,s) a transition from state $\mathbf{j}$ to state i_c in channel c occurs for the observation $\mathbf{O}^{(r)}_{t_0,s,t_1,c}$, and zero otherwise. The estimated mean $\tilde{\mu}^{s,k,c}_{j,m}$, the covariance $\tilde{\mathbf{U}}^{s,k,c}_{j,m}$ of the Gaussian mixture, and the mixture coefficients $\tilde{w}^{s,k,c}_{j,m}$ for mixture m of state j in super state k are obtained as follows:

   $$\tilde{\mu}^{s,k,c}_{j,m} = \frac{\sum_{r,t_0,t_1}\psi^{(r)}_{t_0,t_1}(s,k,c,j,m)\mathbf{O}^{(r)}_{t_0,s,t_1,c}}{\sum_{r,t_0,t_1}\psi^{(r)}_{t_0,t_1}(s,k,c,j,m)}$$

   $$\tilde{\mathbf{U}}^{s,k,c}_{j,m} =$$

   $$\frac{\sum_{r,t_0,t_1}\psi^{(r)}_{t_0,t_1}(s,k,c,j,m)(\mathbf{O}^{r}_{t_0,s,t_1,c}-\tilde{\mu}^{s,k,c}_{jm})(\mathbf{O}^{(r)}_{t_0,s,t_1,c}-\tilde{\mu}^{s,k,c}_{j,m})^T}{\sum_{r,t_0,t_1}\psi^{(r)}_{t_0,t_1}(s,k,c,j,m)}$$

   $$\tilde{w}^{s,k,c}_{j,m} = \frac{\sum_{r,t_0,t_1}\psi^{(r)}_{t_0,t_1}(s,k,c,j,m)}{\sum_{r,t_0,t_1}\sum_{m=1}^{M}\psi^{(r)}_{t_0,t_1}(s,k,c,j,m)}$$

   where $\psi^{(r)}_{t_0,t_1}(s,k,c,j,m)$ is equal to one if the observation $\mathbf{O}^{(r)}_{t_0,s,t_1,c}$, is assigned to super state k in super channel s, state j in channel c and mixture component m, and zero otherwise.
4. If the observation likelihood computed with the Viterbi algorithm at consecutive iteration is smaller than a specified threshold, then the iteration stops and the parameters of the trained model are saved. Otherwise, steps 2-4 are repeated.

7. FACE RECOGNITION

With the parameters of one EBN trained for each face in the database, the face recognition begins with the extraction of the observation vectors from a test face image, as described in Section 4. Then, the likelihood of optimal state segmentation is computed (Section 5) for the test observation sequence given each of the trained models. Finally, the highest matching score between the observation sequence and the trained models reveals the identity of the test image. In our experiments all HMM and CHMM in the lower layer have six states, and three states per channel respectively. For the ”parent” layer the HMM and CHMM have five states per channel. All states are modeled using a mixture of three Gaussian pdf with diagonal covariance matrix. In order to reduce the computational complexity of the EBN, all CHMM used in our experiments have two channels. We have tested the EBN-based face recognition system on the Georgia Tech database [16]. The database consists of 50 people with 15 face images available for each person. For most of the people the pictures were taken in two or three sessions over a period of three months, allowing for strong variation in size, facial expression, illumination, and rotation in both the image plane and perpendicular to the image plane. Table 2 compares the recognition rates obtained in our experiments training each EBN with 10 images per person and testing on the remaining five. On the same database the face recognition system based on the ”eigenface” method [3] achieved 68% correct recognition. The above table indicates that the CHMM-HMM structures achieve the highest recognition rate, but the relatively small amount of data and the small improvement over the ECHMM cannot draw a clear conclusion as to which of these models is best for face recognition. However, our results show that using a CHMM in the parent layer can improve the recognition rate of the EBNs that use a HMM in the ”parent” layer, as in addition to the flexibility of the latter, the first structures can better model rotations in the image plane. The computational complexity of the models used in this paper is dominated by the calculation of the observation probabilities which is similar in all models. However, for models that include CHMM with larger number of channels the computational complexity is determined by the Viterbi algorithm as described in Table 1.

Model	Recognition Rate
EHMM	87.0%
HMM-CHMM	89.0%
CHMM-HMM	92.2%
ECHMM	91.5%

Table 2: A comparison of the face recognition rates for the EHMM, HMM-CHMM, CHMM-HMM and ECHMM

8. CONCLUSIONS

The EBN, represent a novel statistical model with several application in the analysis and modeling of data with N dimensional dependencies. In this paper we describe a training and recognition algorithm for the EBN derived from the optimal state segmentation. In particular an EBN with two layers can be applied to image recognition problems such as face recognition or character recognition where data is essentially two-dimensional. Our experimental results in face recognition show that the members of the EBN family described in this paper outperform some of the existing approaches such as the eigenface method and the embedded HMM method (also a member of the EBN family). The EBN for faces inherits the flexibility of the HMM and CHMM in terms of natural face variations, scaling, and rotations, while significantly reducing the complexity of the fully connected 2D HMM. In addition, as explained in Section 5 the likelihood of the optimal states segmentation for EBN can be efficiently implemented on parallel machines. We consider that the success of the EBN in face recognition and their general representation can inspire more applications of EBN in other areas of image recognition and in general pattern recognition.

9. 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] R. Chellappa, C. Wilson, and S. Sirohey, “Human and machine recognition of faces: A survey,” *Proceedings of IEEE*, vol. 83, May 1995.

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

[4] A. Pentland, B. Moghaddam, and T. Starner, “View based and modular eigenspaces for face recognition,” in *Proceedings on IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 84–91, 1994.

[5] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs Fisherfaces: Recognition using class specific linear projection,” in *Proceedings of Fourth Europeean Conference on Computer Vision, ECCV’96*, pp. 45–58, April 1996.

[6] 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.

[7] 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.

[8] J. Phillips, “Matching pursuit filters applied to face identification,” *IEEE Transactions on Image Processing*, vol. 7, pp. 1150–1164, August 1998.

[9] R. Brunelli and T. Poggio, “Face recognition:features versus templates,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 15, October 1993.

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

[11] 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.

[12] J. Lia, A. Najmi, and R. Gray, “Image classification by a two-dimensional hidden markov model,” *IEEE Transactions on Signal Processing*, vol. 48, pp. 517–533, February 2000.

[13] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden Markov models for complex action recognition,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 994–999, 1997.

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

[15] A. V. Nefian, L. Liang, X. Pi, X. Liu, and C. Mao, “An coupled hidden Markov model for audio-visual speech recognition,” in *International Conference on Acoustics, Speech and Signal Processing*, 2002.

[16] Georgia Tech Face Database, `ftp://ftp.ee.gatech.edu/pub/users/hayes/facedb/`.