Nonparametric Discriminant HMM and Application to Facial Expression Recognition

Lifeng Shang and Kwok-Ping Chan
Department of Computer Science
The University of Hong Kong, Hong Kong
{lfshang, kpchan}@cs.hku.hk

Abstract

This paper presents a nonparametric discriminant HMM and applies it to facial expression recognition. In the proposed HMM, we introduce an effective nonparametric output probability estimation method to increase the discrimination ability at both hidden state level and class level. The proposed method uses a nonparametric adaptive kernel to utilize information from all classes and improve the discrimination at class level. The discrimination between hidden states is increased by defining membership coefficients which associate each reference vector with hidden states. The adaption of such coefficients is obtained by the Expectation Maximization (EM) method. Furthermore, we present a general formula for the estimation of output probability, which provides a way to develop new HMMs. Finally, we evaluate the performance of the proposed method on the CMU expression database and compare it with other nonparametric HMMs.

1. Introduction

HMMs have already been extensively studied in applications to speech [21], facial expression [20], gesture recognition [12], etc. The discrimination ability of HMMs is often improved by using some discriminative criterions (such as Generalized Probability Descent method [22], Maximum Mutual Information [3], Maximum / Soft Margin [1] and Maximum Minimum Margin [15] [19]) instead of the Maximum Likelihood (ML) to learn model parameters. It has been proven that HMMs trained by such discriminative criterions significantly outperform the traditional non-discriminative HMMs. However, the estimation of model parameters by discriminant criterions has to be converted into other problems first. The authors of [19] converted large margin estimation into a Semi-definite Programming (SDP) problem. In [1], parameters estimation is formulated as a Quadratic Programming (QP) problem. It is a fact that solving a large-scale SDP (or QP) problem is still expensive.

Besides learning with discriminative criterions, the performance of HMMs can also be improved by introducing new state output probability estimation methods, such as Artificial Neural Networks [24], Wavelet [8] and Kernel methods [12]. Recently, Lefevre [18] defined a nonparametric probability density function by k-nearest neighbors (k-NN). The k-NN estimation attempted to introduce discrimination at state level. Their method only utilized information from single class to train model parameters, thus the discrimination at class level is not increased. In [16], another widely used nonparametric density estimation method, Parzen Windows [9], is used to estimate the output probability. Their method avoided the estimation of the mean and covariance compared to Mixture Gaussian (MixG), but discrimination at class level or at state level was not considered. These existing works motivated us to define a new effective nonparametric output probability estimation method to increase the discrimination both at state level and class level.

Inspired by [18], we associate each reference vector with hidden states by membership coefficients, which act as the posterior probabilities of reference vectors belonging to hidden states. These coefficients are further used to estimate the observation probabilities of reference vectors, thus the discrimination at state level is increased. At the learning stage, the adaption of such coefficients is given by the "Baum-Welch algorithm" (BWA) [21]. To improve the discrimination ability at class level, unlike existing works that use some discriminative criterions, we introduce the Linear Interpolation with Maximum Entropy (LIME) density estimation method [13] into output probability estimation. The LIME is a probability estimation method from view of Maximum Entropy [14], which improves the accuracy of probability estimation without assumptions on prior distributions. Since the information of all classes is used in LIME, the discrimination at class level is improved. Furthermore, we present a general formula for the output probability estima-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

tion, which provides a way to develop new HMMs. Some existing estimation methods (e.g. MixG, discrete HMMs, etc) can be derived as a special case of the general formula. Finally, we evaluate the performance of the proposed method on the CMU expression database [17].

The rest of this paper is organized as follows. Section 2 gives an overview of HMM and LIME. In Section 3, we describe the proposed LIME / HMM system. In Section 4, we apply the proposed method to facial expression recognition and its performance is evaluated by the CMU facial expression Database. Section 5 summarizes this paper.

2. HMM and LIME

In this section, we will give a brief review for HMM and LIME to establish notations.

2.1. HMM

HMM consists of nodes representing hidden states, interconnected by links governing the transitions between the states [9]. Each hidden state is also associated with an output probability distribution. In the follows, we give a brief review on HMM. A detailed tutorial on HMM can be found in [21].

A HMM is characterized by the following parameters:

1. The hidden states $S = \{S_1, S_2, \cdots, S_N\}$, where N is the number of states.

2. The state transition probability distribution $A = \{a_{i,j}\}$, where $a_{i,j} = P(q_{t+1} = S_j | q_t = S_i)$, $1 \leq i, j \leq N$, and q_t is the state at time t.

3. The output probability distribution $B = \{b_i(O_t)\}$, where $b_i(O_t) = P(O_t | q_t = S_i)$, $1 \leq t \leq T$, $1 \leq i \leq N$, T is the length of an observation sequence.

4. The initial state distribution $\pi = \{\pi_i\}$, where $\pi_i = P(q_1 = S_i)$, $1 \leq i \leq N$.

HMM is often indicated by the compact notation $\lambda = (A, B, \pi)$. Before applying HMM to a real world problem, the following three central problems have to be solved:

1. Evaluation problem. Find $P(O|\lambda)$ for the observation sequence $O = O_1 O_2 \cdots O_T$.

2. Decoding problem. Find the sate sequence $Q = q_1 q_2 \cdots q_T$ that best explains the observation O.

3. Learning problem. Find $\lambda^* = \arg\max_\lambda P(O|\lambda)$.

The evaluation problem is solved by the "Forward algorithm" [21]. "Viterbi algorithm" [21] solves the Decoding problem. The parameters of HMM are iteratively adapted by a generalized EM algorithm BWA [21].

2.2. LIME

LIME is a nonparametric, adaptive kernel estimation method, and the resulting kernel is asymmetric [13].

Let $\Gamma = \{X_i | X_i \in \mathbb{R}^d, i = 1, \cdots, n\}$ denote the training feature vectors and let $x \in \mathbb{R}^d$ denote observation vector. The objective of LIME is to produce an n-dimensional weight vector $w = \{w_1, w_2, \cdots, w_n\}$ (also called LIME weights and $\sum_{i=1}^n w_i = 1$) from Γ and x. The LIME weights is computed as

$$w^* = \arg\min_w \left(D\left(\sum_{i \in J} w_i X_i - x \right) + \lambda \sum_{i \in J} w_i \ln w_i \right), \tag{1}$$

where function $D(\cdot)$ is a continuous convex function, J is the indices of k-NN of x from the training set Γ, and the parameter λ specifies a tradeoff between reproduction distortion and maximum entropy. LIME uses the linear interpolation equations to avoid bias and the maximum entropy principle to weight all near neighbors as uniformly as possible to keep estimation variance low. Following [13], the mean squared error is used for D and w^* is optimized by a fast primal-dual log-barrier interior-point method.

Through restating LIME minimization in a form of Jaynes's maximum entropy estimates [14], if $i \in J$, the LIME weights are given by

$$w_i^* = \frac{e^{-a^T(X_i - x)}}{\sum_{j \in J} e^{-a^T(X_j - x)}}, \tag{2}$$

where w_i^* is i-th entry of set w^*, a is the d-dimensional Lagrange multiplier. It can be seen that the LIME weights can be expressed in terms of an adaptive kernel and can be used for density estimation.

In the next section, we integrate LIME with HMM to extend its application to sequential data classification. This integration also improves the discrimination ability of HMM at class level, since the information of all classes are used in the optimization of LIME weights.

3. LIME / HMM model

In this section, we discuss the proposed HMM and present a general output probability formula.

3.1. Output probability definition

In our method, LIME and k-NN are integrated in the expression of the state output probability

$$b_i^c(O_t) = \frac{\sum_{j=1}^{M^c} P(S_i^c | x_j^c) \times w(O_t, x_j^c)}{\sum_{j=1}^{M^c} P(S_i^c | x_j^c)}, \tag{3}$$

where O_t is the observation at time t, S_i^c is the i-th hidden state of class c, x_j^c is the j-th reference vector of class c, $P(S_i^c | x_j^c)$ is the posterior probability of x_j^c belonging to the i-th hidden state, $w(O_t, x_j^c)$ is the LIME weight between the observation O_t and the reference vector x_j^c, M^c is the number of reference vectors of class c.

If there is no prior knowledge on reference vectors, we can assume the prior probabilities $P(x_i^c) \equiv P(x_j^c)$ for any $1 \leq i, j \leq M^c$ and the equation (3) can be rewritten as

$$\begin{aligned} b_i^c(O_t) &= \sum_{j=1}^{M^c} P(x_j^c|S_i^c)P(O_t|x_j^c) \\ &= \sum_{j=1}^{M^c} P(O_t, x_j^c|S_i^c) = P(O_t|S_i^c), \end{aligned} \quad (4)$$

where the probability $P(x_j^c|S_i^c)$ is estimated by k-NN

$$P(x_j^c|S_i^c) = \frac{P(S_i^c|x_j^c)}{\sum_{j=1}^{M^c} P(S_i^c|x_j^c)} \quad (5)$$

and the LIME weight $w(O_t, x_j^c)$ acts as the likelihood of x_j^c with respect to O_t. The equation (4) explains why the formula (3) can be used to estimate the state output probability and how the discrimination at state level is improved by k-NN. In the follows, we will give the computation of LIME weights and model parameters and explain how the discrimination at class level is increased.

3.2. Computing LIME weights

To compute the LIME weights $w(O_t, x_j^c)$, we first select the k-NNs of O_t from training samples. Let J^c represent the indices of k-NNs of O_t from the training samples of class c. Similar to the equation (1), if $j \in J^c$, $w(O_t, x_j^c)$ is computed by solving the following optimization problem

$$\begin{aligned} \text{Minimize:} \quad & D\left(\sum_c \sum_{j \in J^c} w(O_t, x_j^c)x_j^c - O_t\right) \\ & +\lambda \sum_c \sum_{j \in J^c} w(O_t, x_j^c)\ln(w(O_t, x_j^c)) \\ \text{Subject to:} \quad & \sum_c \sum_{j \in J^c} w(O_t, x_j^c) = 1, \end{aligned}$$

otherwise $w(O_t, x_j^c) \equiv 0$. $w(O_t, x_j^c)$ measures the similarity between O_t and x_j^c, and by the exemplar based model the posterior probability of the observation O_t classified to class c is

$$P(c|O_t) = \sum_{j=1}^{M^c} w(O_t, x_j^c). \quad (6)$$

From (3) and (6), it can be seen the formula (3) is a weighted version of the posterior probability $P(c|O_t)$ with $P(x_j^c|S_i^c)$ acting as the weighting factors, which explains how the discrimination at the class level is improved.

3.3. Training model parameters

In the learning phase, the model parameters $\lambda^c = (A^c, B^c, \pi^c)$ are learned to maximize the probability of the observation sequence given the model, $P(O|\lambda^c)$. The well-known iterative BWA is used for the estimation of λ^c. We will first establish notations used.

The probability of being in state i at time t given the observation sequence O and the model λ^c is $\gamma_t^c(i)$. The estimate for the parameters A^c and π^c is identical to that given in [21]. The iterative formula for the posterior probability $P(S_i^c|x_j^c)$ is

$$\overline{P}(S_i^c|x_j^c) = \frac{\sum_{t=1}^{T} \gamma_t^c(i,j)}{\sum_{t=1}^{T} \sum_{i=1}^{N^c} \gamma_t^c(i,j)}, \quad (7)$$

where

$$\gamma_t^c(i,j) = \gamma_t^c(i) \cdot \frac{P(S_i^c|x_j^c) \times w(O_t, x_j^c)}{\sum_{j=1}^{M^c} P(S_i^c|x_j^c) \times w(O_t, x_j^c)} \quad (8)$$

is the probability of being in state i at time t with the j-th reference vector accounting for O_t.

From equation (7), the estimate of probability $P(x_j^c|S_i^c)$ is

$$\overline{P}(x_j^c|S_i^c) = \frac{\overline{P}(S_i^c|x_j^c)}{\sum_{j=1}^{M^c} \overline{P}(S_i^c|x_j^c)} = \frac{\sum_{t=1}^{T} \gamma_t^c(i,j)}{\sum_{t=1}^{T} \sum_{j=1}^{M^c} \gamma_t^c(i,j)} \quad (9)$$

which is just the standard estimate of mixture coefficients for continuous HMMs except that x_j^c are used as the reference vectors instead of mean vectors, and this confirms the convergence of the BWA using the estimate method of $P(S_i^c|x_j^c)$ given in (7).

Since the optimization surface usually has many local maxima, the BWA leads to local maxima only. Thus, the initialization of model parameters $P(S_i^c|x_j^c)$ is very crucial. It is achieved by the well-known Fuzzy C-mean (FCM) clustering algorithm in this work. The number of clusters is the same as that of hidden states.

3.4. Relationship to existing works

From equation (4), it can be observed that the proposed nonparametric output probability expression can be written in a more general formula

$$b_i^c(O_t) = \sum_{j=1}^{M^c} P(x_j^c|S_i^c) \times k(O_t, x_j^c), \quad (10)$$

where $k(O_t, x_j^c) \in [0, 1]$ is a kernel function measuring the similarity between the observation O_t and the reference vector x_j^c and $P(x_j^c|S_i)$ is the observation probability of the j-th reference vector in state i. Different HMMs can be obtained by different estimate methods for the observation probabilities of reference vectors $P(x_j^c|S_i)$ or by different definitions for $k(O_t, x_j^c)$.

For the traditional continuous HMM, the output probability is commonly a MixG

$$b_i^c(O_t) = \sum_{j=1}^{M^c} m_{ij}^c \times \phi(O_t; \mu_{ij}^c, \Sigma_{ij}^c), \tag{11}$$

where M^c is the number of mixtures, m_{ij}^c is the mixture coefficient for the j-th mixture in state i and $\sum_{j=1}^{M^c} m_{ij} = 1$, ϕ is typically a Gaussian with mean μ_{ij}^c and covariance Σ_{ij}^c. The output probability (11) can be attributed to the formula (10) with the mean vector μ_{ij}^c acting as the reference vector, defining $P(\mu_{ij}^c|S_i^c) \triangleq m_{ij}^c$ and $k(O_t, \mu_{ij}^c) \triangleq \phi(O_t; \mu_{ij}, \Sigma_{ij})$.

Similarly, for the traditional discrete HMMs, reference vectors are the codevectors $V^c = \{v_1^c, v_2^c, \cdots, v_{M^c}^c\}$ and kernel function can be simply defined as

$$k(O_t, v_j^c) \triangleq \begin{cases} 1, & \text{if } v_j^c \text{ is the nearest neighbour of } O_t \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

In [18], the probability $P(x_j^c|S_i^c)$ is estimated by

$$P(x_j^c|S_i^c) \triangleq \frac{u_i(x_j^c)}{\sum_{j=1}^{M^c} u_i(x_j^c)}, \tag{13}$$

where $u_i(x_j^c)$ is the membership coefficient defined as in Fuzzy set theory

$$u_i(x_j^c) \in (0,1), \quad \sum_{i=1}^{N^c} u_i(x_j^c) = 1 \tag{14}$$

and $k(O_t, x_j^c)$ is computed by

$$k(O_t, x_j^c) \triangleq \begin{cases} 1, & \text{if } x_j^c \text{ is a k-NN of } O_t \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

In [16], $P(x_j^c|S_i^c)$ is estimated by another nonparametric density estimation method-Parzen Windows

$$P(x_j^c|S_i^c) \triangleq \sum_{k=1}^{M^c} \xi_i(x_k^c) \times \psi_h(x_k^c - x_j^c), \tag{16}$$

where ψ_h is a kernel function with bandwidth h and $\xi_i(x_k^c)$ is the traditional discrete HMM observation probability.

The formula (10) gives some insight to define new output probabilities through designing various probabilities $P(x_j^c|S_i^c)$ (k-NN, Parzen Windows, RBF Neural Network, Wavelet, etc), similarity functions $k(O_t, x_j^c)$ (Gaussian, LIME weight, etc.) or different reference vectors (representative exemplars, training samples, codevectors, etc.). In the next section, we will show that the performance of the methods in [16] and [18] can be improved by integrating the proposed nonparametric kernel method or by using the adaption formula (7) and holding their own kernel functions.

4. Facial Expression Recognition

Facial expression recognition has a variety of applications in human computer interface, image retrieval and data-driven animation, etc. Most of the existing facial expression recognition methods attempt to recognize six prototypic expressions (namely, joy, surprise, anger, disgust, sadness and fear) proposed by Ekman [10]. Over the past decades, many techniques (Neural networks [23], Support Vector Machines [2], Local Parameterized Models [4], etc.) have been introduced into still facial image recognition.

Recently, the methods on facial expression have been moving to model the dynamics of facial deformation by integrating temporal information, which is typically handled using Dynamic Programming (DP) or HMMs. Otsuka et al. [20] were the first to apply continuous left-to-right HMMs to recognizing sequences of emotion. To represent the variation in facial expression among persons, they chose a MixG density for approximating the output probability. Yeasin et al. [25] used the discrete HMMs to model temporal facial expression signatures produced by k-NN classifiers. In [6], Cohen et al. proposed a multilevel HMMs, in which the state sequences of the first level HMMs were used as the input of the higher level HMM, for segmenting and recognizing human facial expression. In this section, we apply the proposed nonparametric discriminant HMM to facial expression recognition and evaluate its performance on the CMU facial expression database.

4.1. Feature Extraction and Indication

Feature extraction is the basis for any recognition system, and the features should realistically describe the physical phenomena. In facial expression recognition, there are two types of facial features: permanent and transient features. The permanent facial features are the shapes and locations of eyebrows, eyes lids, nose, lips and chin. The transient features are the wrinkles and bulges appeared with expressions. In this paper, we use the movement of permanent facial features away from neutral positions to measure facial expression variation.

We applied the well-known Active Appearance Model (AAM) [7] on facial image sequences to track the movement of facial features. Figure 1(a) shows the shape model consisting of 58 facial points which is identical with the one given in [5]. Figure 2 displays the facial feature localization results of the subject's six basic expressions.

Based on the facial action code system (FACS) [11], it can be found that the movements of some facial points (e.g. facial points 1 and 13) are not so important to measuring facial deformation. Thus, a subset is selected from these 58 facial points depicted in Fig. 1(b), where the solid triangles and rectangles represent only the X or Y-coordinates are used as features and the solid circles represent both X

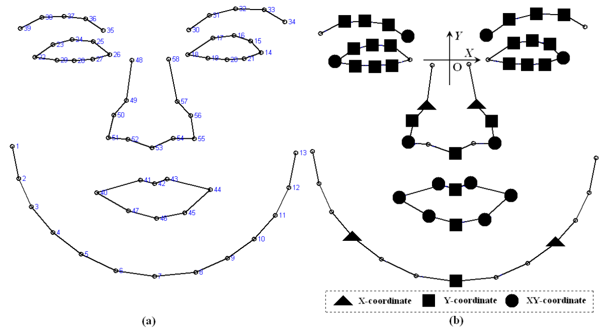

Figure 1. (a)The facial landmarks(58 facial points) and (b) selected feature points.

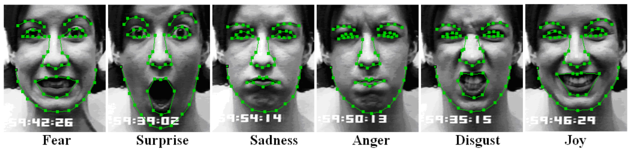

Figure 2. The tracking results of one subject's six basic expressions.

and Y-coordinates are used as features. The midpoint of inner corners of two eyes (facial points 18 and 26) is defined as the coordinate origin. Each frame of video to be recognized is represented by a 52-dimensional feature vector.

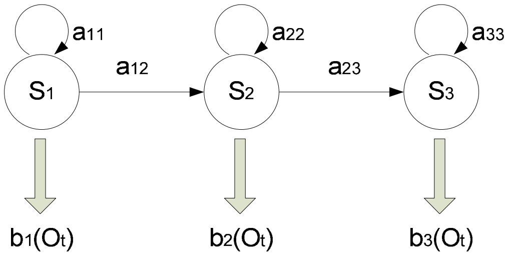

Figure 3. Left-Right HMM.

We employed a left-to-right HMM, where a transition is allowed only to the right-neighbor state or itself as in Figure 3 to model the spatial-temporal variation of facial expression. The model consists of three hidden states S_1, S_2 and S_3, which correspond to the Neutral, Transient and Apex stages of the evolution of an expression. The output probability is defined in (3) and adapted by (7).

4.2. Experimental results and Evaluation

We use the CMU Database to evaluate the performance of the proposed nonparametric discriminant HMM. This database consists of 100 university students ranging in age from 18 to 30 years. Sixty-five percent were female, fifteen percent were African-American and three percent Asian or Latino. For our experiment, we selected 72 whole image sequences (totally, 1085 images) from the database. Each expression contains 12 sequences. The original frames are normalized to 170 × 210 pixels facial images based on the positions of two eyes.

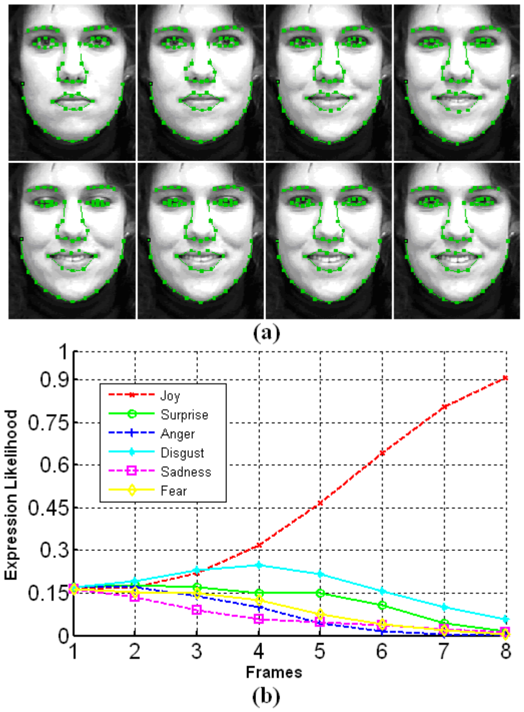

Figure 4. Example 1: (a)An image sequence shows a subject performing smiling, (b) the recognition result.

We will first use two examples to illustrate the efficiency of our method in an intuitive way. In the first example, we created a short image sequence as shown in Figure 4(a) in which the subject performs smiling. We can observe that starting from the fourth frame, lip corners begin to be pulled obliquely and cheeks are raised. Fig. 4(b) presents the expression likelihood probabilities for the six basic expressions. From this figure, the six expressions are close in likelihood at the beginning three frames, which implies that these three frames have Neutral expression. As the expression progresses with time, the likelihood of joy increases gradually. This experimental result illustrates that our method can well model the evolution of facial expression.

Figure 5(a) shows another image sequence in which the subject showed surprise with some frames mis-tracked. In frames 4 and 6, we can see that the locations of mouth and chin are tracked erroneously. Fig. 5(b) gives the result of our method. From this figure, we can observe that although the likelihood of surprise visibly decreases in the sixth frame because of the tracking error, the facial expression can still be correctly recognized. This example illustrates that our method is robust with respect to tracking error.

Table 1. Comparison with different HMM-based methods

	JOY	SUR	ANG	DIS	SAD	FEA	Overall
[16]	91.67	91.67	100.00	58.33	100.00	75.00	86.11
[16]*	91.67	91.67	100.00	75.00	100.00	83.33	90.28
[18]	75.00	100.00	91.67	100.00	100.00	83.33	91.67
[18]*	91.67	100.00	100.00	91.67	100.00	91.67	95.83
Our	100.00	100.00	100.00	91.67	100.00	91.67	97.22

Table 2. Comparison with different HMM-based methods (Small Training dataset)

	JOY	SUR	ANG	DIS	SAD	FEA	Overall
[16]	83.33	100.00	100.00	79.17	95.83	83.33	90.28
[18]	75.00	100.00	83.33	95.83	100.00	79.17	88.89
Our	95.83	100.00	87.50	91.67	100.00	91.67	94.44

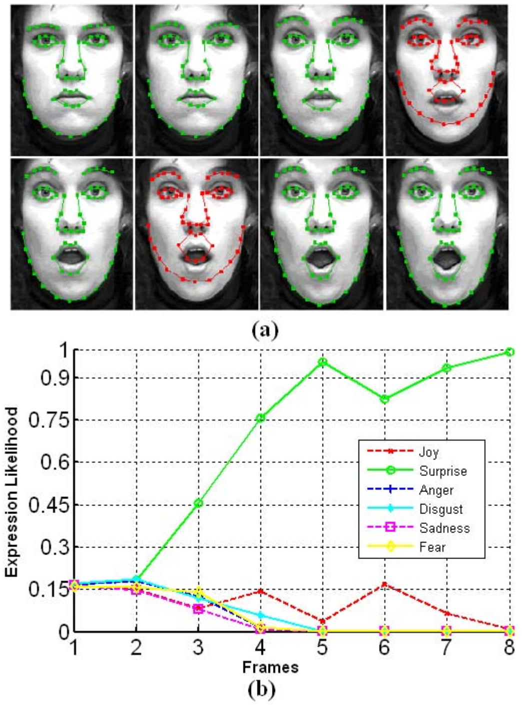

Figure 5. Example 2: (a)An image sequence shows a subject performing surprise with tracking error in the frames 4 and 6, (b) the recognition result.

Furthermore, we used a three-fold cross validation in our experiments to verify the benefit of improving discrimination ability at both state and class levels. Table 1 presents the recognition results of our method, methods in [16], [18] and their corresponding improvements [16]*, [18]* based on the general formula (10). All experiments were performed on the same data set, two folders of image sequences were used as training set and the remaining image sequences were used as testing set. The proposed method achieves a 97.22 percent overall recognition rate and outperforms the other four nonparametric HMM methods. To increase the discrimination ability of the method in [16] in state level, we used (7) to adapt the variation of $\xi_i(x_k^c)$ in method [16]*. The recognition rate increases from 86.11 percent to 90.28 percent. This shows the benefit of increasing discrimination between hidden states. In method [18]*, we integrated LIME with method in [18] by replacing (15) with LIME weights to increase the discrimination at class level. We can see that the recognition rate increases from 91.67 percent to 95.83 percent. This illustrates the importance of improving discrimination ability at class level. Furthermore, we can see that our method still outperforms the improved method [18]*. This confirms the benefit of the adaption formula(7), since the only difference between our method and method [18]* is the adaption of membership coefficients.

To evaluate the performance of our method on small training dataset, we use one folder of facial image sequences as training dataset (each expression has only 4 sequences) and the remaining two folders of sequences as testing dataset. Recognition results summarized in Table 2 illustrate that out method still outperforms the methods in [16] and [18]. Both the method in [16] and our method outperforms the method in [18], since large size of training dataset is essential for k-NN probability estimation.

Compared with our method and the method in [18], the method in [16] is the fastest, since it does not need to find k-NN. Refinement in terms of neighborhood selection and fast neighbor search will be considered in future work to obtain better performance.

5. Conclusion

This paper proposed a new nonparametric HMM for facial expression recognition. We introduced LIME and membership coefficients to HMM, which increased the discrimination ability at both class level and state level. Furthermore, we presented a general formula for output probability estimation, which provides a way to develop new HMM. Experiments on CMU expression database confirmed the efficiency of the proposed method in modeling the evolution of facial deformation. Moreover, the experimental results showed that the performance of some existing HMMs can be improved by integrating the proposed nonparametric kernel method and parameters adaption formula.

References

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *Proc ICML*, 1–8, 2003.

[2] M.S. Bartlett, G. Littlewort, I. Fasel, and J.R. Movellan. Real time face detection and expression recognition: development and application to human-computer interaction. In *CVPR Workshop on CVPR for HCI*, 2003.

[3] Y. Bengio, R. D. Mori, G. Flammia, and R. Kompe. Global optimization of a neural network-hidden markov model hybrid. *IEEE Transactions on Neural Networks*, 3(2):252–259, Mar. 1992.

[4] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[5] Y. Chang, C. Hu, and M. Turk. Probabilistic expression analysis on manifolds. In *Proc IEEE CVPR*, 520–527, June 2004.

[6] I. Cohen, A. Garg, and T. Huang. Emotion recognition from facial expressions using multilevel HMM. In *NIPS Workshop on Affective Computing*, 2000.

[7] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.

[8] L. Couvreur and C. Couvreur. Wavelet-based method for nonparametric estimation of HMMs. *IEEE Signal Processing Letters*, 7(2):1–3, Feb. 1999.

[9] R. O. Duda, P. E. Hart and D. G. Stork. *Pattern classification*. John Wiley Interscience, Oct. 2001.

[10] P. Ekman and R. Davidson. *The nature of emotion: fundamental questions*. Oxford University Press, New York, 1994.

[11] P. Ekman and W.V. Friesen. *Facial action coding system (FACS): manual.* Palo Alto, Calif: Consulting Psychologists Press, 1978.

[12] A. Elgammal, V. Shet, Y. Yacoob, and L.S. Davis. Learning dynamics for exemplar-based gesture recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 571–578, June 2003.

[13] M. R. Gupta, R. M. Gray, and R. A. Olshen. Nonparametric supervised learning by linear interpolation with maximum entropy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):766–781, May 2006.

[14] E. T. Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70(9):939–952, Sept. 1982.

[15] H. Jiang, X. Li, and C. Liu. Large margin hidden markov models for speech recognition. *IEEE Trans. Audio, Speech, Lang. Process.*, 14(5): 1584–1595, Sep. 2006.

[16] N. Jin and F. Mokhtarian. A non-parametric HMM learning method for shape dynamics with application to human motion recognition. In *Proc IEEE ICPR*, 29–32, 2006.

[17] T. Kanade, J.F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 46–53, 2000.

[18] F. Lefevre. Non parametric probability estimation for HMM-based automatic speech recognition. *Computer Speech and Language*, 17(2-3):113–136, April-July 2003.

[19] X. Li and H. Jiang. Solving large-margin midden markov model estimation via semidefinite programming. *IEEE Trans. Audio, Speech, Lang. Process.*, 15(8): 2383–2392, Nov. 2007.

[20] T. Otsuka and J. Ohya. Recognizing multiple persons' facial expressions using HMM based on automatic extraction of significant frames from image sequences. In *Proc IEEE ICIP*, 546–549, 1997.

[21] L. R. Rabiner. A tutorial on hidden markov models and selected application in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb. 1989.

[22] N. Thakoor and J. Gao. Shape classifier based on generalized probabilistic descent method with hidden Markov descriptor. In *Proc IEEE ICCV*, 495–502, Oct. 2005.

[23] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, Feb. 2001.

[24] E. Trentin and M. Gori. A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing*, 37(1-4):91–126, April 2001.

[25] M. Yeasin, B. Bullot, and R. Sharma. From facial expression to level of interest: a spatio-temporal approach. In *Proc IEEE CVPR*, 922–927, June 2004.