EMOTION RECOGNITION FROM SPEECH VIA BOOSTED GAUSSIAN MIXTURE MODELS

Hao Tang[1]*, Stephen M. Chu*[2]*, Mark Hasegawa-Johnson*[1]*, Thomas S. Huang*[1]

[1]Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, Urbana, I.L. 61801, USA
[2]IBM T. J. Watson Research Center, Yorktown Heights, N.Y. 10598, USA

ABSTRACT

Gaussian mixture models (GMMs) and the minimum error rate classifier (i.e. Bayesian optimal classifier) are popular and effective tools for speech emotion recognition. Typically, GMMs are used to model the class-conditional distributions of acoustic features and their parameters are estimated by the expectation maximization (EM) algorithm based on a training data set. Then, classification is performed to minimize the classification error w.r.t. the estimated class-conditional distributions. We call this method the EM-GMM algorithm. In this paper, we introduce a boosting algorithm for reliably and accurately estimating the class-conditional GMMs. The resulting algorithm is named the Boosted-GMM algorithm. Our speech emotion recognition experiments show that the emotion recognition rates are effectively and significantly "boosted" by the Boosted-GMM algorithm as compared to the EM-GMM algorithm. This is due to the fact that the boosting algorithm can lead to more accurate estimates of the class-conditional GMMs, namely the class-conditional distributions of acoustic features.

***Index Terms*—** Emotion recognition, Gaussian mixture model, Bayesian optimal classifier, EM algorithm, boosting

1. INTRODUCTION

Speech emotion recognition is a relatively new direction in the areas of speech signal processing and pattern recognition [1–13]. Unlike speech recognition, which aims to extract the linguistic content from a speech signal while considering the emotions carried in the signal as irrelevant noise, speech emotion recognition aims to extract the non-lexical, paralinguistic information from the speech signal regardless of its verbal content. Like speech recognition, speech emotion recognition has turned out to be an important research topic and has many useful applications in our daily lives [3, 4].

Just like speech recognition and many other pattern recognition problems, the problem of speech emotion recognition is often tackled by generative model-based pattern recognition methods such as Gaussian mixture models (GMMs) [14] and hidden Markov models (HMMs) [15] as well as through classification of low-level acoustic features such as mel-frequency cesptral coefficients (MFCCs) or perceptual linear prediction (PLP) coefficients [16, 17]. What is different is that, because speech emotion recognition normally requires text-independency, ignoring the linguistic content of the speech signals, the temporal characteristics or time order of the speech signals are not considered. In practice, a common approach is to utilize a GMM to represent the probabilistic distribution of acoustic features extracted from all speech signals that carry a particular category of emotion in a training data set, and to perform minimum error rate (MER) classification based on the trained emotion-specific GMMs using the Bayesian optimal classifier [18]. The very nice property of GMMs that they can approximate arbitrarily complex probabilistic distributions arbitrarily closely makes GMMs a popular choice for modeling the class-conditional probability distribution functions (PDFs) in many pattern recognition problems.

A GMM is typically estimated by the expectation maximization (EM) algorithm [19] or its variants. One known problem of maximum likelihood estimation techniques such as the EM algorithm is that the estimate is not globally optimal. Most often, the EM algorithm gets stuck in the local maximum of the data log likelihood function. This problem can become very severe when bad initializations of the model parameters are given. Since the accuracy of model estimation is the most important, or even deciding, factor for generative model-based classification, how to reliably and precisely estimate the class-conditional GMMs based on a training data set becomes a central problem of speech emotion recognition.

In this paper, we introduce a novel algorithm for estimating the class-conditional GMMs based on a boosting framework. The algorithm is named Boosted-GMM and can be deemed as an example of the increasingly popular ensemble methods for data analysis [20]. The theoretical study of Mason et. al. [21] showed that boosting can be viewed as gradient descent search in a function space. Rosset et. al. [22] applied this methodology to probability density estimation. Wang et. al. [23] specialized it for GMMs. In this paper, we extend this idea and apply the proposed Boosted-GMM algorithm to speech emotion recognition. Our experiments

This work was funded in part by the U.S. Government VACE Program and in part by the NSF Grant CCF 04-26627.

show that significantly higher emotion recognition rates are achieved by the Boosted-GMM algorithm than are achieved by the EM-GMM algorithm under the same training and test conditions. This result is primarily due to the fact that the boosting algorithm can lead to more accurate estimates of the class-conditional GMMs, namely the class-conditional distributions of acoustic features.

2. DATABASE DESCRIPTION

We have collected a database of emotional speech for analysis and synthesis tasks. Our script consists of 720 semantically-neutral English sentences which were chosen to maximize the phonetic coverage. A student actress whose mother language is American English was hired to speak each of these sentences, as naturally as possible, in the neutral, happy, sad, and angry manners, respectively. The speech waveforms were recorded in a studio environment at 44.1K Hz using a MOTU 8pre firewire audio interface and a Studio Projects B1 condenser microphone, and were downsampled to 16K Hz prior to further processing. The average length of the utterances in the database is about 3 to 4 seconds, depending on the emotion category. Thus, each of the four emotion categories in the database contains 720 utterances, that is speech data about 36 to 48 minutes long.

3. GMM AND MER CLASSIFIER

The GMM [14] confines the form of the PDF to be a linear superposition of a finite number of Gaussian distributions

$$p(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu_k}, \Sigma_k) \tag{1}$$

where α_k is the mixture weight of the k^{th} component Gaussian of the form

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu_k}, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_k})^T \Sigma_k^{-1} (\mathbf{x}-\boldsymbol{\mu_k})} \tag{2}$$

with a mean vector $\boldsymbol{\mu_k}$and covariance matrix Σ_k for a D-dimensional random variable $\mathbf{x}$. α_k can be interpreted as the *a priori* probability that an observation of $\mathbf{x}$ comes from the source governed by the k^{th} Gaussian distribution. Thus it satisfies the constraints $0 \leq \alpha_k \leq 1$ and $\sum_{k=1}^{K} \alpha_k = 1$. A GMM is completely specified by its parameters $\boldsymbol{\theta} = \{\alpha_k, \boldsymbol{\mu_k}, \Sigma_k\}_{k=1}^{K}$ and the estimation of the PDF reduces to finding the proper values of $\boldsymbol{\theta}$based on a training data set. In the context of C-class classification, a class-conditional GMM is trained for each of the C classes $p(\mathbf{x}|c), c = 1, ..., C$ where c denotes the class label. The minimum error rate (MER) or Bayesian optimal classification rule [18] is given by

$$H(\mathbf{x}) = \underset{c \in \{1,...,C\}}{\arg\max}\ p(c|\mathbf{x}) = \underset{c \in \{1,...,C\}}{\arg\max}\ p(\mathbf{x}|c)p(c) \tag{3}$$

where $p(c)$ is the prior probability of the c^{th} class.

A central problem of GMM-MER classification is how to estimate the model parameters $\boldsymbol{\theta_c}$, $c = 1, ..., C$. This problem can be practically solved by maximum likelihood estimation (MLE) techniques such as the EM algorithm. A fundamental drawback of MLE is that it suffers from the local maximum problem, especially when there is insufficient training data. The number of free parameters of a GMM, N, depends on the feature dimension D and the number of Gaussian mixtures K. More precisely, $N = KD^2/2 + 3KD/2 + K - 1$, which grows linearly in K and quadratically in D. In order to alleviate this “curse of dimensionality”, diagonal covariance matrices are often used instead of full covariance matrices in the component Gaussians. In this case, $N = 2KD + K - 1$, which grows linearly in both K and D.

4. THE F-J ALGORITHM

The choice of the number of mixtures, K, is a fundamental issue of GMM training with the EM algorithm. In most cases, K is empirically set prior to training and held fixed afterwards. Problems arise when an inappropriate value of K is assumed. With too big a K, the estimated PDF may overfit the training data, while with too small a K, the estimated PDF can be a very poor approximation of the underlying “true” distribution of the data. The Figueiredo-Jain (F-J) algorithm [24] is an improved variant of the EM algorithm that aims to overcome this weakness. It starts with an arbitrarily large number of mixture components and adjusts it during the estimation process by annihilating those components not supported by the data. The objective function that the F-J algorithm minimizes is

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, Y) &= \frac{N_c}{2} \sum_{k:\alpha_k>0} \log(n\alpha_k) + \frac{k_{nz}}{2}(1 + \log\frac{n}{12}) \\ &\quad - \log p(Y|\boldsymbol{\theta}) \end{aligned} \tag{4}$$

where N_c is the number of parameters needed to specify a mixture component, $p(Y|\boldsymbol{\theta}) = \prod_{\mathbf{y} \in Y} p(\mathbf{y}|\boldsymbol{\theta})$ is the data likelihood, and $k_{nz} = \sum_k [\alpha_k > 0]$ denotes the number of non-zero-probability components. A component-wise EM algorithm [25] is adopted to minimize Eq. 4, which leads to the following iterative update formulas

$$\begin{aligned} \omega_{k,i} &= \frac{\alpha_k^t p(\mathbf{y_i}|k, \boldsymbol{\theta^t})}{\sum_{j=1}^{K} \alpha_j^t p(\mathbf{y_i}|j, \boldsymbol{\theta^t})} \qquad (5) \\ \alpha_k^{t+1} &= \frac{\max\left\{0, (\sum_{i=1}^{n} \omega_{k,i}) - \frac{N_c}{2}\right\}}{\sum_{j=1}^{K} \max\left\{0, (\sum_{i=1}^{n} \omega_{j,i}) - \frac{N_c}{2}\right\}} \\ \boldsymbol{\mu_k}^{t+1} &= \frac{\sum_{i=1}^{n} \mathbf{y_i} \omega_{k,i}}{\sum_{i=1}^{n} \omega_{k,i}} \\ \Sigma_k^{t+1} &= \frac{\sum_{i=1}^{n} \omega_{k,i} (\mathbf{y_i} - \boldsymbol{\mu_k}^{t+1})(\mathbf{y_i} - \boldsymbol{\mu_k}^{t+1})^T}{\sum_{i=1}^{n} \omega_{k,i}} \end{aligned}$$

5. THE BOOSTED-GMM ALGORITHM

The goal of GMM model estimation (or model estimation in a very general sense) is to seek a set of model parameters that maximizes the data log likelihood. Given a training data set $X = \{\mathbf{x}_i\}_{i=1}^N$ and a probability density function $p(\mathbf{x})$ to be estimated, the data log likelihood is given by

$$L(p) = \sum_{i=1}^{N} \log p(\mathbf{x}_i) \tag{6}$$

Here, in this paper, $p(\mathbf{x})$ is the probability density function of a GMM given by Equation 1. Instead of directly optimizing Equation 6 as in the EM algorithm, we start with an initial estimate p_0 (a GMM) and iteratively add to this estimate a small component q_t at round t. That is,

$$p_t = (1-\alpha)p_{t-1} + \alpha q_t \tag{7}$$

where $0 \leq \alpha \leq 1$ is small quantity and q_t is also a GMM. According to Taylor's theorem, the new data log likelihood

$$L(p_t) = L((1-\alpha)p_{t-1} + \alpha q_t) \tag{8}$$

can be approximated by a first-order Taylor expansion around p_{t-1}, namely

$$L(p_t) = L(p_{t-1}) + N\log(1-\alpha) + \frac{\alpha}{1-\alpha}\sum_{i=1}^{N}\frac{q_t(\mathbf{x}_i)}{p_{t-1}(\mathbf{x}_i)} \tag{9}$$

Equation 9 implies that, in order to maximize the data log likelihood, we can first search for $q_t \in Q$ where Q is the space of GMMs such that

$$q_t = \arg\max_{q_t \in Q} \sum_{i=1}^{N} \frac{q_t(\mathbf{x}_i)}{p_{t-1}(\mathbf{x}_i)} \tag{10}$$

Then, given the q_t, we can seek the α that yields maximum increase in $L(p_t)$. From Equation 10, it is obvious that q_t can be obtained through performing maximum likelihood estimation on the training examples weighted by $W_t = \frac{1}{p_{t-1}}$. This meets our intuition of boosting that more focus is put on the examples with low probabilities under the previous estimate, and W_t can be deemed as the distribution over the training set at round t in a boosting algorithm [26]. The Boosted-GMM algorithm is summarized in Algorithm 1.

The sampling procedure in Step 3 in Algorithm 1 can be done as follows. At each round, we sort the training examples by their weights in the descending order and keep only a fraction r of them (e.g. $r = 0.3$). Another heuristic is, at each round, to keep the first N_t examples where

$$N_t = \mathit{Round}\left(e^{-\sum_{i=1}^{N} W_t(\mathbf{x}_\mathrm{i})\log W_t(\mathbf{x}_\mathrm{i})}\right) \tag{11}$$

Finally, once all class-conditional GMMs are estimated, the MER or Bayesian classifier is given by

$$k(X) = \arg\max_{1\leq k\leq K} p(X|k)p(k) \tag{12}$$

Algorithm 1 The Boosted-GMM algorithm

```
1: Input: $X = \{\mathbf{x}_i\}_{i=1}^N$, $r$, and $T$.
2: Initialize $W_1(\mathbf{x}_i) = 1/N$, $i = 1, ..., N$, $p_0 = 0$.
3: For $t = 1, ..., T$ or until $L(p_t) \leq L(p_{t-1})$
       • Sample $X_t$ from $X$ according to $W_t$ and estimate $q_t$ from $X_t$ using the F-J algorithm [24].
       • Set $p_t = (1-\alpha)p_{t-1} + \alpha p_t$ where $\alpha = \arg\max_{0\leq\alpha\leq 1} L(p_t)$.
       • Update $W_{t+1}(\mathbf{x}_i) = \frac{1}{p_t(\mathbf{x}_i)}$, $i = 1, ..., N$.
4: Output: Final density estimate $p_T$.
```

where $p(X|k) = \prod_{i=1}^{n} p(\mathbf{x}_i|k)$ is the likelihood of a test utterance with n speech frames, $X = \{\mathbf{x}_i\}_{i=1}^n$, and $p(k)$ is the prior probability of the k^{th} class (i.e. emotion category).

6. EXPERIMENTS

In this paper, we performed speech emotion recognition experiments on the emotional speech database described in Section 2. For each experiment, we randomly selected from the database a training set consisting of 10 utterances per emotion and a test set consisting of 90 utterances per emotion. Therefore, the training set consisted of 40 utterances in total and the test set 360 utterances in total. Note that there were no overlapping utterances in the training and test sets. Instead of conducting a complete cross-validation process, which would be very time consuming, we ran 100 such experiments independently, each of which involved a random selection of the training set and test set from the database, and the emotion recognition rates of these 100 experiments were averaged. We believed that in this way such average would represent a well generalized emotion recognition rate over the database. An experiment was carried out as follows. For each speech frame in an utterance, we extracted a set of acoustic features including 12 MFCCs, the log energy, and the pitch (f_0) using a 25ms hamming window at a 10ms frame rate. These basic parameters were augmented with their first derivative to form for each frame a 28-dimensional feature vector. Based on the training set, the class-conditional feature vector distributions were estimated using both the Boosted-GMM algorithm and the EM algorithm, and the same MER classifier was applied with two sets of estimated class-conditional GMM models.

Figure 1 shows the average overall emotion recognition rates (i.e., the average recognition rates across all 4 emotions of 100 independent experiments) of the Boosted-GMM algorithm on the test sets with two sampling fractions r versus the number of boosting iterations, as well as compares these recognition rates with the average overall emotion recognition rate of the EM-GMM algorithm on the same test sets. The number of Gaussian mixtures in the GMMs and the num-

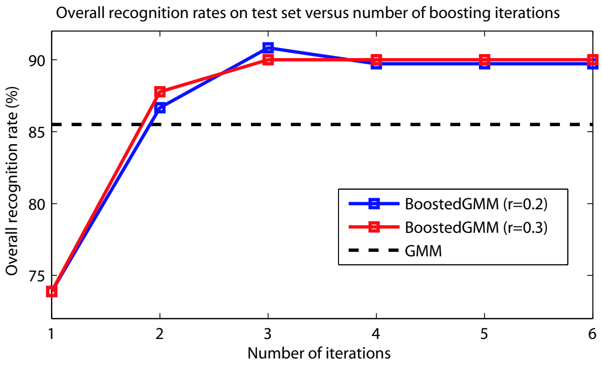

Fig. 1. Comparison of overall emotion recognition rates of the Boosted-GMM algorithm and the EM-GMM algorithm.

ber of EM iterations required for convergence were automatically determined by the F-J algorithm. Our experiment results demonstrate that the emotion recognition rates can be effectively and significantly "boosted" by the Boosted-GMM algorithm, which is a natural, expected result and clear indication that the class-conditional probabilistic density functions can be more accurately estimated by the Boosted-GMM algorithm than by the EM-GMM algorithm. It is worthy to mention that the Boosted-GMM algorithm converges very fast - only a few iterations (less than 5) would be sufficient to lead to a stable result. This relaxes the possible concern that the Boosted-GMM might require a lot more training time than the EM-GMM algorithm.

7. CONCLUSION

In this paper, we introduce the Boosted-GMM algorithm, which embeds the EM algorithm in a boosting framework and which can be used to reliably and accurately estimate the class-conditional probabilistic distributions in any pattern recognition problems based on a training data set. We apply the Boosted-GMM algorithm to speech emotion recognition and our experiments show that the emotion recognition rates are effectively and significantly "boosted" by the Boosted-GMM algorithm as compared to the EM-GMM algorithm due to the fact that boosting can lead to more accurate estimates of the class-conditional GMMs, namely the class-conditional distributions of acoustic features.

8. REFERENCES

[1] Dellaert, F., Polzin, T., Waibel, A., "Recognizing emotion in speech," Proc. ICSLP'96 pp. 1970-1973.

[2] S. Moriyama and S. Ozawa, "Emotion recognition and synthesis system on speech," Proc. ICMCS'99.

[3] Huber, R., Batliner, A., Buckow, J., Noth, E., Warnke, V., Niemann, H., "Recognition of emotion in a realistic dialogue scenario," Proc. ICSLP'00, pp. 665-68.

[4] Petrushin, V., "Emotion recognition in speech signal: experimental study, development, and application," Proc. ICSLP'00.

[5] Oudeyer, P., "Novel Useful Features and Algorithms for the Recognition of Emotions in Human Speech," Proc. ICSP'02.

[6] Schuller R., Rigoll G., Lang M., "Hidden Markov model-based speech emotion recognition," Proc. ICASSP'03, pp. 1-4.

[7] T.L. Nwe, S.W. Foo and L.C. De Silva, "Speech emotion recognition using hidden markov models," Speech Communication 41, (2003), pp. 603-23.

[8] Lee, C.M., Yildirim, S., Bulut, M., Kazemzadeh, A., Busso, C., Deng, Z., Lee, S., Narayanan, S., "Emotion recognition based on phoneme classes," Proc. ICSLP'04.

[9] Dan-Ning Jiang, Lian-Hong Cai, "Speech emotion classification with the combination of statistic features and temporal features," Proc. ICME'04, pp. 1968-1970.

[10] Tao J. H., Kang Y. G., "Features importance analysis for emotional speech classification," in LNCS'05, pp. 449-457.

[11] Chul, M.L. and S. Narayanan, "Toward Detecting Emotions in Spoken Dialogs," IEEE Trans. ASP 13(2):293-303, 2005.

[12] Ververidis, D. and Kotropoulos, C., "Emotional speech recognition: resources, features, and methods," Speech Comm. 48(9):1162-1181, 2006.

[13] D. Neiberg, K. Elenius, and K. Laskowski, "Emotion recognition in spontaneous speech using GMMs," Proc. INTERSPEECH'06.

[14] Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[15] Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, 1989, pp. 257-286.

[16] L. R. Rabiner, B. H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[17] H. Hermansky, "Perceptual linear predictive PLP analysis for speech," JASA, 87(4):1738–1752, 1990.

[18] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern Classification. JohnWiley & Sons, Inc., 2nd edition, 2001.

[19] Arthur Dempster, Nan Laird, and Donald Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. of Royal Stat. Society, B, 39(1):1C38, 1977.

[20] Richard Berk, "An Introduction to Ensemble Methods for Data Analysis" (March 27, 2005). Department of Statistics, UCLA. Department of Statistics Papers. Paper 2005032701. http://repositories.cdlib.org/uclastat/papers/2005032701.

[21] L. Mason, J. Baxter, P. Bartlett and M. Frean, "Boosting algorithms as gradient descent," NIPS 1999.

[22] S. Rosset and E. Segal, "Boosting density estimation," NIPS 2002.

[23] F. Wang, C. Zhang and N. Lu, "Boosting GMM and its two applications," MCS 2005.

[24] M. Figueiredo, A. Jain, "Unsupervised learning of finite mixture models," PAMI 2002.

[25] G. Celeux, S. Chretien, F. Forbes, and A. Mkhadri, "A component-wise EM algorithm for mixtures," Tech. Rep. 674, INRIA, Rhone-Alpes, France.

[26] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. of Computer and System Sciences, no. 55. 1997.