Vol.2010-SLP-84 No.26
2010/12/21

Optimal use of trees in structural MAP adaptation for speaker verification

Sangeeta Biswas,[†1] Marc Ferras,[†1] Koichi Shinoda[†1]
and Sadaoki Furui[†1]

In speaker verification, the Structural Maximum-A-Posteriori(SMAP) adaptation technique is often used to train speaker-adapted acoustic models by using available speech data in an efficient and flexible manner. In SMAP adaptation, a tree structure is used to represent the acoustic space of the human voice. We observed that one particular tree structure is not necessarily optimal for modeling the acoustic space of all speakers. In this paper, we propose a voting approach as a way to combine the decisions of multiple SMAP-adapted systems using different tree structures. We expect that this approach is more robust than using a single tree structure. We evaluate our proposed method on the 10sec4w-10sec4w task of NIST SRE 2006 and show that our method is more effective than the conventional SMAP adaptation as well as relevance MAP adaptation.

1. Introduction

Over the last few years text-independent speaker verification systems have become robust against inter-session variability for speech segments of around 2 or 3 minutes. This is mainly due to the development of the Joint Factor Analysis (JFA)[4] and Nuisance Attribute Projection (NAP)[8] techniques. However, when speech segments are very short, e.g., 10 seconds, the verification accuracy is not as satisfactory as for long segments. In some cases using NAP has been reported to be worse[2] than no compensation.

To tackle this problem, Vogt et al.[9] proposed using Probabilistic Subspace Adaptation (PSA) into Factor Analysis (FA) modeling. Fauve et al.[2] proposed a well-tuned speech detection front-end for improved frame selection followed by eigenvoice modeling. All these methods emphasize keeping the number of model parameters small enough so that they can be reliably estimated.

In speech recognition, Shinoda et al.[7] showed that the Structural Maximum-A-Posteriori (SMAP) adaptation technique using a tree structure performs better than relevance MAP[3] when the size of adaptation data is very small. In speaker verification, Liu et al.[5] and Xiang et al.[10] successfully used it for speech segments of about 2 minutes long or shorter. However, one particular tree structure is not always optimal for modeling the acoustic space of every speaker. In this paper, we propose a voting approach as a way to combine decisions of multiple systems with different tree structures. We expect that this approach is more robust than SMAP adaptation with a single tree structure.

The remainder of the paper is organized as follows. In Section 2, we describe our GMM-SVM based system. A brief description of SMAP adaptation is given in Section 3. Section 4 illustrates our proposed voting approach. In Section 5 and Section 6, we describe our experimental setup and results respectively. Section 7 gives some conclusions.

2. GMM-SVM verification system

The goal of an automatic speaker verification system is to verify the claimed identity of a speaker, giving a binary decision. Adaptation of Gaussian Mixture Models (GMM) was first used for speaker verification by Reynolds et al.[6]. Most speaker verification approaches are currently based on the same framework. Campbell et al.[1] showed that an approach using Support Vector Machines (SVM) and GMM mean vectors as features (GMM-SVM) obtains similar performance to the GMM-UBM paradigm and has less computational complexity.

In a GMM-SVM system, a speaker-independent GMM, so-called Universal Background Model (UBM), is trained using hours of speech segments from hundreds of speakers using the Expectation-Maximization (EM) algorithm. A GMM is obtained for each speech segment of a target speaker by adaptating the UBM. GMM for a set of background speakers, used as negative data in the classifier, are obtained in the same way. The SVM classifies the stacked mean vectors of the speaker models into target (true) or impostor (false) classes.

Let the GMM-UBM have M Gaussian pdf components

†1 Tokyo Institute of Technology

© 2010 Information Processing Society of Japan

$$g(x) = \sum_{i=1}^{M} \lambda_i \mathcal{N}(x; \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \quad (1)$$

where λ_i is a mixture weight, $\mathcal{N}$ is a Gaussian pdf, μ_i and Σ_i are the mean and covariance matrix of the Gaussian pdf respectively and Σ_i is assumed to be diagonal.

For speaker s and the m-th Gaussian, MAP adaptation generates a mean vector as

$$\hat{\boldsymbol{\mu}}_m(s) = \alpha_m \boldsymbol{\mu}_m(s) + (1 - \alpha_m)\boldsymbol{\mu}_m, \quad (2)$$

where $\boldsymbol{\mu}_m(s)$ is the expected value of the m-th Gaussian using the adaptation data only and $\boldsymbol{\mu}_m$ is the corresponding mean vector in the UBM. α_m is a weight used to weight the relevance of the prior and is computed by introducing the so-called *relevance factor* τ as

$$\alpha_m = \frac{\gamma_m}{\gamma_m + \tau}, \quad (3)$$

where γ_m is the occupation count of the m-th Gaussian given the adaptation data.

The SVM classifies so-called *supervectors* obtained by concatenation of the mean vectors of the speaker-adapted models. Prior to classification, the mean vectors are typically normalized by its variance and mixture weight as

$$\mathbf{sv}^s = (\sqrt{\lambda_1}{\boldsymbol{\Sigma_1}}^{-\frac{1}{2}}\boldsymbol{\mu}_1^T, ..., \sqrt{\lambda_M}\boldsymbol{\Sigma}_M^{-\frac{1}{2}}\boldsymbol{\mu}_M^T)^T, \quad (4)$$

where $\mathbf{sv}^s$ has dimension $M \times F$ if the Gaussian mean vectors are F-dimensional.

3. SMAP adaptation

SMAP adaptation was proposed to keep the desirable asymptotic properties of relevance MAP while dealing with the problem of the data scarceness by using a tree structure. First, a tree is obtained by clustering the Gaussian components of the UBM. The root node of the tree represents the whole acoustic space and each of the leaf nodes has a Gaussian component that summarizes its child node distributions. After building the tree, speaker-dependent models are obtained by using each non-leaf node as prior information for its child nodes. These two steps are briefly described in the following two subsections.

3.1 Tree construction

In our method, the structure of the tree represented by the number of layers L and the number of branches $B_r^{(l)}$ from a node r at the l-th layer needs to be provided prior to clustering.

For clustering, we define the distance measure between two Gaussian components as the symmetric Kullback-Leibler (KL) divergence. Assuming the covariance matrices to be diagonal, the KL divergence between two Gaussian components, $g_a(.)$ and $g_b(.)$ can be written as

$$d(a,b) = \sum_{i=1}^{F}[\frac{\sigma_a^2(i) - \sigma_b^2(i) + (\mu_b(i) - \mu_a(i))^2}{\sigma_b^2(i)} + \frac{\sigma_b^2(i) - \sigma_a^2(i) + (\mu_a(i) - \mu_b(i))^2}{\sigma_a^2(i)}], \quad (5)$$

where $\mu_a(i)$ is the i-th element of F-dimensional mean vector μ_a and $\sigma_a^2(i)$ is the i-th diagonal element of covariance matrix $\sum_a$

The algorithm for obtaining a tree from a UBM with G gaussians is given below:

(1) Set:
  (a) the root node to be node k
  (b) all the M Gaussians of UBM in set G_k
  (c) $B_k^{(1)}$ to be the number of children n
  (d) l to be 1

(2) Calculate the node pdf g_k for node k using the following formulas:

$$\mu_k(i) = \frac{1}{M_k}\sum_{m \in G_k} \mu_m(i), \quad (6)$$

$$\sigma_k^2(i) = \frac{1}{M_k}[\sum_{m \in G_k} (\sigma_m^2(i) + \mu_m^2(i)) - M_k\mu_k^2(i)], \quad (7)$$

  where M_k is the number of Gaussian components included in G_k.

(3) If l is equal to L, stop clustering, else go to Step 4

(4) Compute the initial pdf for n child nodes using the *minimax* method:
  (a) Find n Gaussian components from G_k:

© 2010 Information Processing Society of Japan

(i) The 1st Gaussian is $g_{c_1}(.) = g_{\hat{m}}(.)$ where
$$\hat{m} = \arg\max_m d(m, k) \quad (8)$$
(ii) Rest of the $(n-1)$ Gaussians will be $g_{c_p}(.) = g_{\hat{m}}(.)$ where
$$\hat{m} = \arg\max_m \min_{q \in G_{ck}} d(m, c_q), \quad (9)$$
where G_{ck} is the set of Gaussians already assigned to the child nodes of node k, $1 \le p \le n-1$ and $1 \le q \le n-2$

(b) Interpolate the node pdf of the node k and the initial node pdf of each child node c_p and set the node pdf for c_p as follows:
$$\hat{\mu}_{c_p}(i) = (1-\alpha)\mu_k(i) + \alpha\mu_{c_p}(i), \quad (10)$$
$$\hat{\sigma}^2_{c_p}(i) = (1-\alpha)(\sigma^2_k(i) + \mu^2_k(i)) + \alpha(\sigma^2_{c_p}(i) + \mu^2_{c_p}(i)) - \hat{\mu}_{c_p}, \quad (11)$$
where $0 \le \alpha \le 1$.

(5) Repeat the following k-means procedures until the grand sum of distances, $\mathcal{GD}$, converges:
(a) For each Gaussian component in G_k, calculate the distance from it to each child node pdf of the l-th layer by using (5), and assign each mixture component to the nearest child node
(b) Recalculate the child node pdf by using (6) and (7)
(c) Using (5), calculate the sum of distances, $\mathcal{D}$, from each child node to each of its mixture components and then obtain $\mathcal{GD}$ by accumulating all $\mathcal{D}$

(6) Set each child node to be node k and its corresponding subset of Gaussian components to be G_k. Increase l and go to Step 4.

3.2 Adaptation

The formulation of SMAP adaptation is similar to that of relevance MAP[3], except that it uses hierarchical priors and uses normalized pdfs in the formulation. The adaptation steps for each node p using adaptation data $X = \{x_1, x_2, ..., x_T\}$ are:

(1) Transform each sample vector x_t into a vector y_{mt} for each mixture component m as follows:
$$y^{(p)}_{mt} = \mathbf{\Sigma}^{-1/2}_m (x_t - \mu^{(p)}_m), \quad (12)$$
where $t = 1, 2, ..., T$ and $m = 1, 2, ..., M^{(p)}$.

(2) Estimate the normalized pdf $\mathcal{N}(Y^{(p)}|\nu, \eta)$ for $Y^{(p)}_m = \{y^{(p)}_{m1}, y^{(p)}_{m2}, ..., y^{(p)}_{mT}\}$, where $\nu^{(p)}$ and $\eta^{(p)}$ represent the shift and rotation needed to compensate for the distortion, i.e., to adapt the model parameters to the data. When there is no mismatch between the training and adaptation data, then $\nu^{(p)} = \vec{0}$ and $\eta^{(p)} = I$. The ML estimation of the mean vector of the normalized pdf is calculated as follows:
$$\tilde{\nu}^{(p)} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M^{(p)}} \gamma^{(p)}_{mt} y^{(p)}_{mt}}{\sum_{t=1}^{T}\sum_{m=1}^{M} \gamma^{(p)}_{mt}}, \quad (13)$$
where $\gamma^{(p)}_{mt}$ is the occupation probability for Gaussian m at tree node p and time t.

(3) Calculate the hierarchical prior
$$\hat{\nu}^{(p)} = \frac{N_p \tilde{\nu}^{(p)} + \tau \hat{\nu}^{(p-1)}}{N_p + \tau}, \quad (14)$$
where $N_p = \sum_{t=1}^{T}\sum_{m=1}^{M^{(p)}} \gamma(p)_{mt}$ is the average number of frames assigned to node pdf p and τ is the MAP relevance factor that weights the priors at the parent node $p-1$.

(4) Compute the SMAP estimate of the mean vector
$$\hat{\mu}^{(p)}_m = \mu^{(p)}_m + \sum_m^{1/2} \hat{\nu}^{(p)}, \quad (15)$$
where $\mu^{(p)}_m$ is the unadapted mean vector for Gaussian m of node p.

4. Voting method

In SMAP adaptation, a tree structure obtained by clustering Gaussians offers a convenient way to capture the hierarchical structure of the acoustic space of the human voice. Different speakers have different acoustic spaces depending on factors such as the language, accent or pronunciation particularities. It is therefore reasonable to think that the optimal clustering differs from speaker to speaker and, in a hierarchical clustering scenario, so would the optimal tree structure. In other words, some tree structures may be adapted more efficiently to some speakers than others. In the context of SMAP adaptation this would translate into better prior estimates. As evidenced by informal experiments, decisions involving certain speakers are slightly sensitive to the chosen tree structure. In this paper, we propose to combine decisions of multiple systems with different tree structures as a way to mitigate this problem. To proceed, we construct a set of

© 2010 Information Processing Society of Japan

K SMAP adapted systems with different tree structures and:

(1) For each trial x,
  (a) Ask yes/no vote to each of the K systems

$$Vote(x) = \begin{cases} yes & \text{if } thr_k >= score, \\ no & \text{if } thr_k < score, \end{cases} \quad (16)$$

  where thr_k is the speaker independent threshold of system k
  (b) Divide the K systems into two groups based on their types of votes.
  (c) Re-estimate the score by averaging the score of the majority group.
(2) Set a global speaker-independent threshold
(3) Take final decision about each trial as follows

$$Decision(x) = \begin{cases} true & \text{if } thr_G >= score, \\ false & \text{if } thr_G < score, \end{cases} \quad (17)$$

  where thr_G is the global speaker independent threshold

In machine learning, voting and score fusion techniques are known to obtain performance gains if the individual systems are performing sufficiently. Otherwise, the final decisions could degrade and become worse than those of the individual systems.

5. Experimental setup

The performance of the speaker verification systems was measured by carrying out experiments on the 10sec4w-10sec4w task of the 2006 NIST SRE ⋆1. In this task, the length of training and test segments is about 10 seconds. There are 2971 true trials and 30584 false trials for 731 speakers among which 316 are females and 415 are males.

Regarding feature extraction, we first remove the non-speech part from the speech segments using the information in the transcript files. We break each segment into frames of 30 ms long with a frame rate of 100 frames/sec. We pre-emphasize each frame with a pre-emphasis factor of 0.97 and apply a Hamming window. We compute 15 Perceptual Linear Prediction (PLP) coefficients, augmented with the energy coefficient and first and second derivatives, resulting in 48 features per frame. Cepstral mean subtraction was applied to remove static channel effects.

A UBM with 512 Gaussians was trained using about 180 hours of speech involving 2832 and 1974 female and male speech segments of the 2004 NIST SRE. We applied 5 iterations of Baum-Welch re-estimation. For the speaker models we use SMAP adaptation with different relevance factors. The resulting supervectors have 24576 dimensions. We use a soft-margin SVM with a linear kernel. The same 4806 speech segments used for UBM training were used as the imposter speaker data.

6. Results

First we conducted experiments to compare SMAP and relevance MAP in the GMM-SVM system, and later some experiments on the proposed voting approach. We ran one system using relevance MAP and 13 systems using SMAP. The latter used different tree structures that were either binary trees, variations of binary trees or three level trees with the same number of child nodes at every layer. In this series of experiments, both MAP and SMAP systems used a relevance factor of 10.

Table 1 shows the Equal Error Rate (EER) ⋆2 of these systems. Most of the SMAP-adapted systems outperform the relevance MAP-adapted system. The tested binary trees are the worst performing amongst all trees. One reason could be that the number of nodes/clusters is too small to calculate the prior efficiently. That is the reason why as soon as more than two children are included in their layers, e.g. SMAP 2_2_2_5 or SMAP 2_2_5_5, error rates decrease. For the three layer trees, error rates consistently decrease as the number of nodes gets larger. The best relative improvement for individual SMAP systems, around 6%, is obtained for the 2_10_10_2 tree structure-based system, although several other systems perform fairly close. Overall, systems using structures with a larger number of nodes/clusters tend to obtain the lower absolute error rates.

The SMAP voting system outperforms any SMAP individual system, suggesting the voting technique is working properly. We obtain an additional gain of

⋆1 http://www.itl.nist.gov/iad/mig/tests/spk/2006/index.html

⋆2 EER is the rate when False Rejection(FR) error and False Acceptance (FA) error are equal. FA error occurs when a imposter speaker is accepted falsely as the claimed speaker and FR error occurs when a true speaker is rejected against his/her own claim

© 2010 Information Processing Society of Japan

2.3% EER from the best SMAP individual system, which makes a total relative gain of 8.3% EER from the MAP baseline system.

7. Conclusion

We proposed a voting technique to avoid the issue of tree structure optimization in SMAP adaptation. We tested it on a speaker verification task, namely the 10sec4w-10sec4w condition of the 2006 NIST SRE, a inherently difficult task due to the short length of the speech segments. We showed that the voting technique is effective although relative gains are small. We also showed that SMAP-adapted systems outperform a MAP-adapted baseline by a 6% in relative EER terms. As future work, we would investigate other score fusion strategies, e.g., based on neural network or logistic regression. Exploring a greater diversity of trees should be addressed as well.

References

1) Campbell, W.M., Sturim, D.E. and Reynolds, D.A.: Support Vector Machines using GMM supervectors for Speaker Verification, *IEEE Signal Processing Letters*, Vol.13, No.5, pp.308–311 (2006).
2) Fauve, B., N.Evans, N.P., Bonastre, J.F. and Mason, J.: Influence of task duration in text-independent speaker verication, *Proc. Interspeech*, pp.794–797 (2007).
3) Gauvain, J.L. and Lee, C.-H.: Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains, *IEEE Trans. on Speech and Audio Processing*, Vol.2, pp.291–298 (1994).
4) Kenny, P. and Dumouchel, P.: Experiments in Speaker Verification using Factor Analysis Likelihood Ratios, *In Proc. IEEE Odyssey:The Speaker and Language Recognition Workshop*, pp.219–226 (2004).
5) Liu, M., Chang, E. and Dai, B.-Q.: Hierarchical Gaussian Mixture Model for Speaker Verification, *In Proc. ICSLP* (2002).
6) Reynolds, D.A., Quatieri, T.F. and Dunn, R.B.: Speaker Verification using adapted Gaussian Mixture Models, *Digital Signal Processing*, Vol.10, pp.19–41 (2000).
7) Shinoda, K. and Lee, C.-H.: A Structural Bayes Approach to Speaker Adaptation, *IEEE Trans. on Speech and Audio Processing*, Vol.9, No.3, pp.276–287 (2001).
8) Solomonoff, A., Quillen, C. and Campbell, W.M.: Channel Compansetion for SVM Speaker Recognition, *In Proc. IEEE Odyssey:The Speaker and Language Recognition Workshop*, pp.57–62 (2004).
9) Vogt, R., Lustri, C. and Sridharan, S.: Factor Analysis Modelling for Speaker Verification with Short Utterances, *In Proc. IEEE Odyssey:The Speaker and Language Recognition Workshop* (2008).
10) Xiang, B. and Berger, T.: Efficient Text-Independent Speaker Verification with Structural Gaussian Mixture Models and Neural Network, *IEEE Trans. on Speech and Audio Processing*, Vol.11, pp.447–456 (2003).

Table 1 EER for GMM-SVM systems using MAP and SMAP adaptation on the 10sec4w-10sec4w task of 2006 NIST SRE. The design of a tree is written as $n_1\_n_2\_\ldots\_n_l$ where n_l represents the maximum number of child nodes belonging to each node of the l-th layer.

System	# nodes	EER(%)
MAP	-	32.5
SMAP 2_2_2	8	32.7
SMAP 2_2_2_2	16	32.5
SMAP 2_2_2_2_2	32	32.6
SMAP 2_2_2_5	40	31.9
SMAP 2_2_5_5	100	30.6
SMAP 5_5_5	125	30.9
SMAP 2_2_2_2_10	160	31.4
SMAP 2_5_5_5	250	30.9
SMAP 2_15_2_5	300	30.8
SMAP 7_7_7	343	30.8
SMAP 2_10_10_2	400	30.5
SMAP 2_2_15_15	900	30.7
SMAP 11_11_11	1331	30.7
SMAP Voting System		29.8

© 2010 Information Processing Society of Japan