Posterior Weights and Gaussian Selection for Spoken Language Recognition

Kong-Aik Lee, Changhuai You and Haizhou Li
Institute for Infocomm Research,
Agency for Science, Technology and Research (A*STAR), Singapore
E-mail: {kalee, echyou, hli}@i2r.a-star.edu.sg

***Abstract*— This paper investigates the use of the posterior weights of GMMs for spoken language recognition, the goal of which is to determine the language spoken in speech utterances. Since the modeling of distribution is based on the component weights, the number of components in the GMM has to be sufficiently large so as to provide enough degree of freedom. To this end, Gaussian selection technique is applied to speed up the run-time computation. Another problem in using posterior weights is the distance metric. We treat the posterior weights as the probability mass function of a discrete random variable, for which rigorous similarity measures can be easily defined. The proposed language recognition system achieves state-of-the-art performance on the 1996, 2003, 2005 and 2007 National Institute of Standards and Technology (NIST) language recognition tasks.**

I. INTRODUCTION

Spoken language recognition (SLR) refers to the process of determining the language spoken in an utterance [1]. Automatic SLR has shown to be useful in many applications, such as, automatic routing of emergency calls, multilingual conversational systems, and multilingual speech recognition.

Recent advances have shown successful applications of support vector machines (SVM) for spoken language and speaker recognition [2]. In [3], speech utterances are first represented as Gaussian mixture models (GMMs). Supervectors are then formed by concatenating the mean vectors in the GMMs. The task of classifying speech utterances is thereby translated into an equivalent task of classifying the supervectors corresponding to the speech utterances. In this paper, we propose using the posterior weights of GMM for the reason that a less restrictive distance metric can be defined for posterior weights. We show that the proposed method gives equally good accuracy compared to the method of mean supervector for language classification. It also gives competitive performance compared to the state-of-the-art *parallel phone recognition followed by language modeling* (PPRLM) technique [1, 4, 5].

Previous works reported in [6, 7, 8, 9] use similar form of posterior probabilities out of a GMM as features. Modest successes were achieved. In addition to a properly defined distance metric, this paper shows that significant improvement can be obtained with an increased order of GMM. Since the modeling of speaker-specific distribution is now based on the component weights, the number of components in the GMM has to be sufficiently large so as to provide enough degree of freedom in modeling the distribution.

A subtle problem in increasing the number of Gaussians is the computational complexity. To overcome this problem, we propose two methods for handling large-order GMMs: (i) a divide-and-conquer training strategy, and (ii) a Gaussian selection technique for reducing run-time computation. The fast computation method leads to 10 time faster computation on language recognition tasks as we shall see in Section IV.

II. THE POSTERIOR-WEIGHT VECTOR

A. The Posterior Weights of GMM

The acoustic cepstral distribution of speech has often been explicitly and accurately modeled by GMMs, which can be written in the following form

$$p(\mathbf{x}) = \sum_{i=1}^{M} \pi_i \mathcal{N}(\mathbf{x} \mid \theta_i), \tag{1}$$

where M denotes the number of component Gaussians. The GMM is governed by the set of parameters $\Theta = \{\pi_1, \ldots, \pi_M, \theta_1, \ldots, \theta_M\}$, where π_i are the weights of the mixture components, each of which is parameterized by $\theta_i = \{\boldsymbol{\mu}_i, \mathbf{C}_i\}$ consisting of a mean vector $\boldsymbol{\mu}_i$ and a covariance matrix $\mathbf{C}_i$. Intuitively, the mean, covariance and weight indicate the location, shape and height of a Gaussian component in the GMM.

Let $\Theta^{\text{prior}} = \{\pi_1, \ldots, \pi_M, \theta_1, \ldots, \theta_M\}$ be the parameters of a *world model* representing a language-independent distribution covering all possible languages in the acoustic space (we shall come back to the issue of training the world model in Section III). Given a speech sample $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ of size N, consider re-estimating the weights of the GMM such that the resulting model $\Theta^X = \{\tilde{\pi}_1, \ldots, \tilde{\pi}_M, \theta_1, \ldots, \theta_M\}$ describes better the distribution that has generated X. Using the maximum-likelihood (ML) criterion [10], the posterior weights are given by

$$\tilde{\pi}_i = \frac{1}{N} \sum_{n=1}^{N} P\left(i \mid \mathbf{x}_n, \Theta^{\text{prior}}\right), \text{ for } i = 1, 2, \ldots, M, \tag{2}$$

where

$$P\left(i \mid \mathbf{x}_n, \Theta^{\text{prior}}\right) \equiv \frac{\pi_i \mathcal{N}\left(\mathbf{x}_n \mid \theta_i\right)}{\sum_{j=1}^{M} \pi_j \mathcal{N}\left(\mathbf{x}_n \mid \theta_j\right)} \tag{3}$$

is the posterior probability for each observation $\mathbf{x}_n$.

The prior weights π_i of the world model Θ^{prior} represent our existing knowledge about the heights (i.e., probabilities) of individual Gaussian components in a language-independent distribution. Given that X has been observed, the posterior weights $\tilde{\pi}_i$ represent the heights of the Gaussian components that best describe, in ML sense, the distribution that has generated X.

Given two speech samples, X_a and X_b, we could represent them using (2) and (3) as sets of posterior weights, $\{\tilde{\pi}_i(X_a)\}_{i=1}^{M}$ and $\{\tilde{\pi}_i(X_b)\}_{i=1}^{M}$, respectively. Since the weights of a GMM are non-negative and always sum to one, the component index i of the GMM can be treated as a discrete random variable. We refer to this random variable as the acoustic event, and there are M possible acoustic events with probabilities $P(i) = \pi_i$ for $i \in \{1,2,\ldots,M\}$. With this interpretation, the two speech samples can be further represented as probability mass functions:

$$P_a(i) = \tilde{\pi}_i(X_a) \text{ and } P_b(i) = \tilde{\pi}_i(X_b). \tag{4}$$

The similarity between X_a and X_b can now be measured using P_a and P_b (assumed to be independent) as the expected likelihood of one distribution under the other:

$$E_a\{P_b(i)\} = E_b\{P_a(i)\} = \sum_{i=1}^{M} P_a(i) P_b(i). \tag{5}$$

It should be mentioned that the above similarity measure is appropriate, as in our case, as long as the mean vectors and covariance matrices of the world model are held fixed during the ML re-estimation.

B. Using the Posterior Weights with SVM

Stacking the posterior weights in (2), we form an M-dimensional vector $\boldsymbol{\rho}$ representing the observation sequence X. The transformation from X to $\boldsymbol{\rho}$ can be cast as a nonlinear mapping, as follows

$$X \mapsto \boldsymbol{\rho}(X) = [\tilde{\pi}_1, \tilde{\pi}_2, \ldots, \tilde{\pi}_M]^T, \tag{6}$$

where the superscript T denotes matrix transposition. The dimension M of the mapping can be made sufficiently high (by using a larger world model) such that $M \gg D$, where D is the dimension of the acoustic feature space, thus, (6) is equivalent to a nonlinear mapping onto a higher dimensional vector space.

Using (5) and (6), the similarity between two speech samples, X_a and X_b, is given by

$$K(X_a, X_b) = [\boldsymbol{\rho}(X_a)]^T \boldsymbol{\Lambda}^{-1} [\boldsymbol{\rho}(X_b)], \tag{7}$$

where $\boldsymbol{\rho}(X_a)$ and $\boldsymbol{\rho}(X_b)$ are the posterior-weight vectors, and $\boldsymbol{\Lambda}$ is a square positive-definite matrix for normalization purpose. In this paper, $\boldsymbol{\Lambda}$ is taken as the within-class covariance matrix [11]. This inner-product form of similarity measure is of particular usefulness for SVM. It allows an SVM to measure the similarity between two variable-length sequences as the inner product between the posterior-weight vectors having the same dimension M.

C. Language Recognition System using Posterior-Weight Vector and SVM

At the training phase, training samples from all target languages are first transformed into posterior-weight vectors as in (6) and normalized using $\boldsymbol{\Lambda}^{-1/2}$. Each of these vectors is then assigned with an appropriate label (i.e., +1 for target language, -1 for competing languages) for SVM training using a *one-versus-rest* strategy. Repeating the training procedure for other target languages (by taking other languages as competing languages) results in a set of language-dependent SVM models:

$$\begin{aligned} f_k(\boldsymbol{\rho}) &= \sum_{l=1}^{L_k} \alpha_l y_l K\left(X_{l,k}, X\right) + \beta_k \\ &= \underbrace{\left(\sum_{l=1}^{L_k} \alpha_l y_l \boldsymbol{\rho}_{l,k}\right)^T}_{\mathbf{w}_k} \boldsymbol{\rho} + \beta_k \end{aligned} \tag{8}$$

for $k = 1,2,\ldots,K$, where K denotes the number of target languages, L_k is the number of support vectors, β_k is the bias parameter, and α_l is the weights assigned to the support vector $\boldsymbol{\rho}_{l,k}$ with its label given by $y_l \in \{-1,+1\}$. The support vectors, $\boldsymbol{\rho}_{1,k}, \boldsymbol{\rho}_{2,k}, \ldots, \boldsymbol{\rho}_{L_k,k}$, can be combined to form a compact model $\mathbf{w}_k$ as shown in (8).

For a given test sample, we compute and normalize the posterior-weight vector following the same procedure as in the training phase. The posterior-weight vector is then scored against each model by simply computing the inner product $\mathbf{w}_k^T \boldsymbol{\rho}$ followed by a shifting of β_k.

III. Fast Posterior-Weight Computation

Gaussian selection (GS) methods are commonly used in speech recognition to speed up state-likelihood computation [12]. In this section, we use a divide-and-conquer strategy in training the world model and propose a GS method for this type of composite model. The run-time computation is further reduced with tied-covariance modeling.

A. A Composite World-Model

Consider that we have access to a collection of speech samples in B number of languages. We could train a GMM for each language l_b in the following form

$$p(\mathbf{x} \mid l_b) = \sum_{q=1}^{Q} \pi_{q,b} \mathcal{N}\left(\mathbf{x} \mid \theta_{q,b}\right), \text{ for } b = 1,2,\ldots,B, \tag{9}$$

where Q denotes the number of mixture components and $\pi_{q,b}$ is the weight of the qth Gaussian component characterized by

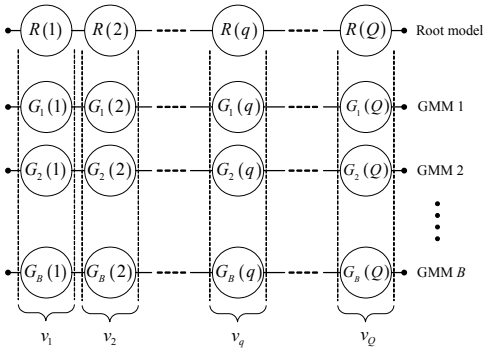

Fig. 1 Each component Gaussian $R(q)$ in the root model is associated with a subset (or shortlist) v_q of Gaussian components, one from each of the B adapted GMMs.

parameters $\theta_{q,b}$. Pooling the B GMMs with equal priors, we arrive at

$$p(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} p(\mathbf{x}\,|\,l_b) = \frac{1}{B}\sum_{b=1}^{B}\sum_{q=1}^{Q} \pi_{q,b}\mathcal{N}\left(\mathbf{x}\,|\,\theta_{q,b}\right), \qquad (10)$$

where $B \times Q = M$ is the number of components in the resulting composite model. Comparing (10) to (1), the world model is now formed by pooling B language-dependent GMMs instead of training one GMM from the pooled data.

All the B datasets (one for each language) jointly contribute to the formation of a composite world-model with $M = B \times Q$ components. We refer to these languages as the basis languages to distinguish them from the target languages mentioned earlier in Section II. Such divide-and-conquer strategy is effective in dealing with the data imbalance problem. Furthermore, training smaller GMM using smaller subpopulation is easier than training a large GMM using all data. This strategy also leads to a fast posterior-weight computation technique as presented next.

B. Gaussian Selection

For a large GMM, an input vector will be near only to a few Gaussian components, while lying on the tail of most components (thus lower likelihood). Gaussian selection (GS) method attempts to efficiently select and only compute the likelihood of a subset (or shortlist) of Gaussian components in the vicinity of each input vector.

Consider that the language-dependent GMMs $p(\mathbf{x}\,|\,l_b)$ in (9) are trained from a common root-model. (The expectation-maximization (EM) algorithm is well suited for this task). Gaussian components in the adapted GMMs would therefore retain a one-to-one correspondence with that of the root model. This relationship is represented as

$$v_q = \{G_b(q)\}_{b=1}^{B},\ q = 1,2,\ldots,Q, \qquad (11)$$

where v_q is the shortlist that associates a Gaussian component $R(q)$ in the root model with B Gaussian components, one from each of the adapted GMMs. We use $R(q)$ and $G_b(q)$ to indicate the Gaussian component q of the root model and adapted model, respectively. As illustrated in Fig. 1, there are in total Q shortlist, each contains B member Gaussians.

For a given input vector, we find τ number of Gaussian with highest likelihood using the root model. For the case of tied-covariance (i.e., all component Gaussians share the same covariance matrix), this is equivalent to finding the set of τ Gaussians, ϕ, with smallest distance $\left\|\mathbf{x}'_n - \mu'_{R(q)}\right\|^2$ from the input vector:

$$\phi = \arg\min_{q}(\tau)\left\{\left\|\mathbf{x}'_n - \mu'_{R(q)}\right\|^2\right\} \qquad (12)$$

where $\mathbf{x}'_n$ and $\mu'_{R(q)}$ are variance-normalized input and mean (of the root model) vectors. The final shortlist is given by the union of shortlists associated with the top-scoring Gaussians:

$$v = \bigcup_{q\in\phi} v_q\ . \qquad (13)$$

The likelihoods of the Gaussians in the final shortlist v are evaluated, while the remaining Gaussians are assumed to have zero likelihood. This method leads to $Q/(Q/B+\tau)$ times faster computation. The parameter τ is typically smaller than Q by at least one order of magnitude.

IV. EXPERIMENTS

We evaluate the performance of the proposed method following the framework established by NIST [13]. NIST evaluation treats language recognition as a detection problem rather than identification.

The experiments are conducted on the 1996, 2003, 2005 and 2007 NIST LRE databases. There are twelve target languages for LRE 1996 and LRE 2003, and seven for LRE 2005 (marked with *). These target languages include Arabic, English*, Farsi, French, German, Hindi*, Japanese*, Korean*, Mandarin*, Spanish*, Tamil* and Vietnamese. The most recent LRE 2007 involves fourteen target languages, namely, Arabic, Bengali, Chinese (comprised of Mandarin and three dialects), English, Farsi, German, Hindustani (comprised of Hindi and Urdu), Japanese, Korean, Russian, Spanish, Tamil, Thai and Vietnamese. We evaluate the performance in terms of the equal-error-rate (EER) computed from the pooled set of 30-second trial scores.

The training data is drawn primarily from the CallFriend database. All the development and test data are pre-processed by a speech activity detector to remove silence. Shifted-delta-cepstral (SDC) coefficients are then formed using Mel-frequency cepstral coefficients with (7, 1, 3, 7) configuration resulting in feature vectors of dimension 49 [4].

We compare four approaches, namely, (i) SVM with posterior-weight vector (PW-SVM), (ii) SVM with GMM mean supervector (GSV-SVM) [3], (iii) GMM [4] and (iv) PPRLM [5]. For the proposed PW-SVM system, the number of front-end basis languages are fixed at B = 12, each contributing Q = 2,048 Gaussian components. This setting results in an acoustic resolution of M = 24,576. With $\tau = 32$

TABLE I
EER (%) PERFORMANCE OF FOUR SYSTEMS ON THE 1996, 2003, 2005 AND 2007 NIST LREs FOR 30 SECONDS TEST DURATIONS.

Duration	System	1996	2003	2005	2007
30s	PW-SVM	1.99	3.42	5.97	5.42
	GSV-SVM	2.79	4.01	5.91	5.84
	GMM	4.00	4.77	8.30	9.68
	PPRLM	2.14	2.17	4.38	6.51

TABLE II
FUSION RESULTS ON THE 1996, 2003, 2005 AND 2007 NIST LREs FOR 30 SECONDS TEST DURATION. (* THE 1996 LRE DATASET IS USED TO TRAIN THE FUSION WEIGHTS.)

Systems	1996	2003	2005	2007
PW-SVM	1.99	3.42	5.97	5.42
+ GSV-SVM	1.58*	2.77	4.67	4.46
+ GMM	1.81*	2.67	5.11	5.60
+ PPRLM	1.08*	1.67	3.32	4.08
Fuse All	0.94*	1.25	2.85	3.45

the posterior weight computation technique described in Section III leads to approximately 10 times faster computation. For the GSV-SVM system, we used a background model with 512 mixtures. The background model is adapted on a per utterance basis to produce 25,088-dimensional supervectors. We closely follow the system setup in [4] for the GMM system, where each target language is modeled with two gender-dependent GMMs with 2,048 mixtures. The PPRLM uses three phone recognizers (Czech, Hungarian, and Russian) based on long temporal context developed by Brno University of Technology [5]. The phone recognizers are trained on the SpeechDat-East database. There are in total 158 phones, 45 for Czech, 61 for Hungarian and 52 for Russian. We use a setup similar to [5] (i.e., a 1-best PPRLM system) except that the output scores are merged from the individual PRLM subsystems by taking their average.

The performances of individual systems are listed in Table I. It is evident that the performance of the PW-SVM is consistently better than that of the baseline GMM system on the 1996, 2003, 2005 and 2007 LREs. It is also clear that the PW-SVM performs competitively when compared to the state-of-the-art GSV-SVM and PPRLM. Table II shows the fusion results of the PW-SVM with other systems. We use the linear logistic regression fusion implemented in the FoCal toolkit [14] to calibrate and fuse the scores. The fusion of PW-SVM with PPRLM works extremely well considering their diversity in language modeling strategies. The PW-SVM also fuses rather nicely with the GSV-SVM and GMM, even though the improvement is comparatively less than the fusion with PPRLM. Fusing all systems results in significant EER improvements of 53%, 63%, 52% and 36% over the PW-SVM baseline for the 1996, 2003, 2005 and 2007 LREs, respectively.

V. CONCLUSIONS

We have presented a new perspective on representing variable-length speech utterances and measuring their similarity using the posterior weights of GMMs for spoken language recognition. We showed that the posterior weights of GMMs can be seen as probability mass functions of discrete distributions. This interpretation allows reliable similarity measure to be derived in terms of expected likelihood without any undesirable approximation. We also showed that high-order GMM can be easily trained using a divide-and-conquer strategy, and proposed a Gaussian selection technique for the sake of low run-time computation. Language recognition experiments show that the proposed method performs consistently well across the 1996, 2003, 2005 and 2007 NIST language detection tasks. It also fuses nicely with other approaches giving an EER of 0.94 %, 1.25 %, 2.85 % and 3.45 % on the 1996, 2003, 2005 and 2007 NIST LREs, respectively.

REFERENCES

[1] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 1, pp. 31-44, Jan. 1996.

[2] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 210-229, 2006.

[3] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker recognition," *IEEE Signal Processing Lett.*, vol. 13, no. 5, pp. 308-311, May 2006.

[4] E. Singer, P. A Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, "Acoustic, phonetic, and discriminative approaches to automatic language identification," in *Proc. Eurospeech*, pp. 1345-1348, 2003.

[5] P. Matějka, P. Schwarz, J. Černocký, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," in *Proc. Eurospeech*, 2005, pp. 2237-2240.

[6] V. Wan and S. Renels, "Speaker verification using sequence discriminant support vector machines," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 2, pp. 203-210, Mar. 2005.

[7] N. Scheffer and J.-F. Bonastre, "UBM-GMM driven discrimitive approach for speaker verification," in *Proc. Odyssey*, 2006, pp. 1-7.

[8] M. Liu, Z. Zhang, M. Hasegawa-Johnson, and T. S. Huang, "Exploring discriminative learning for text-independent speaker recognition," in *Proc. IEEE ICME*, 2007, pp. 56-59.

[9] K. A. Lee, C. You, and H. Li, "Spoken language recognition using support vector machines with generative from-end," in *Proc. IEEE ICASSP*, 2008, pp. 4153-4156.

[10] X. Huang, A. Acero, H. -W. Hon, *Spoken Language Recognition*: *A Guide to Theory, Algorithm, and System Development*. NJ: Prentice Hall, 2001.

[11] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. INTERSPEECH*, 2006, pp. 1471-1474.

[12] M. J. F. Gales, K. M. Knil, and S. J. Young, "State-based Gaussian selection in large vocabulary continous speech recognition using HMMs," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 2, pp. 152-161, Mar. 1999.

[13] A. F. Martin and J. S. Garofolo, "NIST speech processing evaluations: LVCSR, speaker recognition, language recognition," in *Proc. IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, pp. 1-7, 2007.

[14] N. Brümmer, *FoCal*: *Toolkit for Fusion and Calibration*. Available: http://www.dsp.sun.ac.za/~nbrummer/ focal/.