Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

Bhattacharyya-based GMM-SVM System with Adaptive Relevance Factor for Pair Language Recognition

Chang Huai You[+], Haizhou Li[+], Eliathamby Ambikairajah[++], Kong Aik Lee[+], Bin Ma[+]

[+] Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
[++] School of Electrical Engineering and Telecommunications
The University of New South Wales, Sydney, Australia
{echyou,hli,kalee,mabin}@i2r.a-star.edu.sg, ambi@ee.unsw.edu.au

Abstract

In this paper, we develop a hybrid system for pair language recognition using Gaussian mixture model (GMM) supervector connecting to support vector machine (SVM). The adaptation of relevance factor in maximum *a posteriori* (MAP) adaptation of GMM from universal background model (UBM) is studied. In conventional MAP, relevance factor is empirically given by a constant value. It has been proven that the relevance factor can be dependent to the particular application. We use the relevance factor to control the degree of influence from the observed training data for more effectiveness. In order to design a robust pair language recognition system, we develop a hybrid scheme by using separate-training Bhattacharyya-based kernels with the adaptive relevance factor applied. The pair language recognition system is verified on National Institute of Standards and Technology (NIST) language recognition evaluation (LRE) 2011 task. Experiments show the improvement of the performance brought by the proposed scheme.

Index Terms: maximum *a posteriori*, supervector, Gaussian mixture model, support vector machine

1. Introduction

Language recognition is a speech signal processing to recognize the language of a spoken utterance. The pair language recognition is to detect the language type in the context of a fixed pair of languages. Given a segment of speech and a specified language pair, i.e., two of the possible target languages of interest, the task is to decide which of these two languages is in fact spoken in the given segment. The techniques include the acoustic and phonotactic modelings. The parallel-phone recognition followed by language modeling (PPR-LM) [1] is a classic phonotactic approach using phone tokenistic statistics; while Gaussian mixture model (GMM) related technique is the typical acoustic method. Recently, GMM supervector has been found to achieve state-of-the-art performance in this area.

In this paper, we study on GMM supervector and focus on pair language recognition. GMM supervector SVM is one of the most popular acoustic modeling approaches for its reliable performance [2]. A GMM language model can be trained by using maximum *a posteriori* (MAP) estimation from a universal background model (UBM) [3]. The UBM is usually obtained through expectation-maximization (EM) algorithm from a background dataset covering a sufficiently wide range of languages, speakers, sessions and channels. In MAP, the relevance factor is indirectly affect how much new data could be absorbed to update the parameters (i.e., weight, mean, covariance) of a model. It has been proven that the relevance factor can also be optimized by the particular training data [4]. Conventional MAP does not specify the relevance factor in a systematic manner; in other words, the relevance factor is usually set empirically. Most of researchers like to use an appropriate fix value in place of the data-dependent value. In the GMM-UBM system, the relevance factor is not so sensitive due to the nature of generative modeling [5] and therefore can be fixed. In GMM-SVM language recognition system, a GMM supervector is used to represent the language property of an utterance and serves as an input vector to the SVM. Since SVM works in a discriminative manner [6], the relevance factor could sensitively affect the position, which represents a language, in the supervector space. It is necessary to mitigate the variation of database so that supervectors can well manifest the saliency of language characteristics.

Since we discuss the GMM supervector rather than the GMM probability, the solution of the recognition problem can be sought in the supervector domain. Actually, the supervector deduced from the MAP criterion can be also derived in supervector domain through the probabilistic analysis [4]. In [7], we shew the effectiveness of the adaptation of the relevance factor to the duration of the utterance. In this paper, we develop the pair language recognition system in connection with the adaptive relevance factor.

In the SVM framework, we need to define a kernel to compare supervectors for classification. For pair language recognition, we choose two ways to perform the recognition: one-to-all core-to-pair modeling and one-to-one pair modeling. The two ways are merged into a hybrid system. We propose a separated training scheme for both Bhattacharyya-based mean and covariance kernels [8]. In the hybrid system, we consider the combination of the mean supervector and covariance supervector. The validity of the data-dependent relevance factor will be investigated by using the pair language recognition system on the NIST LRE 2011 30-second task [9].

In the remainder of the paper, we describe the conventional MAP for GMM in section 2. We introduce the relevance factor for MAP estimation in section 3. We develop a hybrid pair language recognition system in section 4.1. The performance evaluation is reported in section 5. We summarize the paper in section 6.

2. MAP for Language Recognition

Given UBM model

$$\mathrm{u} = \{\bar{\omega}_i, \bar{\mathbf{m}}_i, \bar{\Sigma}_i; i = 1, 2, ..., C\} \quad (1)$$

we have the corresponding language-dependent GMM,

$$\lambda = \{\omega_i, \mathbf{m}_i, \Sigma_i; i = 1, 2, ..., C\} \quad (2)$$

where $\mathbf{m}_i, \Sigma_i, \omega_i, (i = 1, ..., C)$ are respectively the mean vector, the covariance matrix, and the weight of the ith Gaussian component. The UBM is trained with EM algorithm using a large dataset covering different languages, channels and speakers to form a language-independent model [5].

For the MAP adaptation to λ, prior probability, $P(\lambda)$, should be given λ. With the MAP criterion, λ is selected such that it maximizes the *a posteriori* probability,

$$\lambda = \arg\max_{\lambda} P(\lambda|\mathbf{X}) = \arg\max_{\lambda} \left[f(\mathbf{X}|\lambda)g(\lambda)\right] \quad (3)$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\kappa]$ is the sequence of feature vectors used to train the GMM, λ; $\mathbf{x}$ is a J-dimensional feature vector; and κ is the number of feature vectors. As a result of (3), the mean parameters of the ith Gaussian are adapted as follows [5],

$$\mathbf{m}_i = \alpha_i \check{\Xi}_i + (1 - \alpha_i)\bar{\mathbf{m}}_i \quad (4)$$

where Ξ_i is the first order sufficient statistics. α_i is the data-dependent adaptation coefficients, which is given by

$$\alpha_i = \frac{N_i}{N_i + \gamma_i} \quad (5)$$

The relevance factor γ_i is the parameter in the normal-Wishart density as which the Gaussian parameters are modeled. However, in conventional MAP, the relevance factor is given as a fixed value, and the occupation rate N_i is theoretically given by

$$N_i = \sum_{t=1}^{\kappa} \frac{\omega_i p(\mathbf{x}_t|\mathbf{m}_i, \Sigma_i)}{\sum_{l=1}^{C} \omega_l p(\mathbf{x}_t|\mathbf{m}_l, \Sigma_l)} \quad (6)$$

where $p(\cdot)$ denotes probabilistic function.

Using a data-dependent adaptation coefficient allows a mixture-dependent adaptation of parameters. If a mixture component has a low occupation rate N_i of new data, then $\alpha_i \to 0$ causing the deemphasis of the new parameters. For mixture components with high probabilistic counts, $\alpha_i \to 1$, causing the use of the new language-dependent parameters. It is obviously found that the relevance factor is a way of controlling how much new data should be observed in a mixture before the new parameters begin replacing the old parameters. Thus, this approach should be robust to sparse training data.

3. Relevance Factor of MAP

MAP can be used to estimate the parameters of the probabilistic distribution. When conventional MAP method is applied for GMM parameter estimation, its relevance factor is considered as fixed value, which is believed not to be an optimal solution. To meet the realistic requirement of the GMM-supervector based description for language recognition, we have to analyze the influence to the relevance factor and to provide a proper solution in the MAP algorithm. In this session, we will show the relationship between the relevance factor and the statistics of the training data.

3.1. Determination of relevance factor

We observed that the supervector deduced from the MAP criterion can be also derived in supervector domain through the probabilistic analysis. We analyze the MAP algorithm from the GMM-supervector perspective. In language recognition, the GMM-supervector is usually generated from UBM to represent the language characteristics according to the related utterances. Here we define the supervector as a concatenation of mean vectors from a GMM. Assume $\mathbf{m}$ represents the UBM supervector and also assume GMM-supervector $\mathbf{m}(\lambda)$ can be constructed by a language independent vector $\bar{\mathbf{m}}$ and a language dependent vector $\tilde{\mathbf{m}}(\lambda) = \Phi\mathbf{z}(\lambda)$, where Φ denotes a transition matrix reflecting some feature of generalized training database and vector $\mathbf{z}(\lambda)$ is related to certain attributes of the particular language. We have

$$\mathbf{m}(\lambda) = \bar{\mathbf{m}} + \Phi\mathbf{z}(\lambda) \quad (7)$$

It is reasonable to assume that Gaussian components in a GMM are independent each other; and further assumption is that the language-dependent vector $\mathbf{z}(\lambda)$ is of the standard normal distribution $\aleph(\mathbf{z}(\lambda)|0, \mathrm{I})$, and Φ is a block diagonal matrix with each block being of dimension $J \times J$, hence the mean vector of the ith Gaussian component can be given by

$$\mathbf{m}_i(\lambda) = \bar{\mathbf{m}}_i + \Phi_i\mathbf{z}_i(\lambda) \quad (8)$$

the natural logarithm of the conditional likelihood function of an observed feature vector $\mathbf{x}$ given the attribute $\mathbf{z}(\lambda)$ is shown below

$$\log P(\mathbf{X}|\mathbf{z}, \lambda) = \Theta + \Omega(\mathbf{z}(\lambda)) \quad (9)$$

where Θ accounts for all terms unrelated to $\mathbf{z}(\lambda)$

$$\Theta = \sum_{i=1}^{C} N_i \log \frac{1}{(2\pi)^{J/2}|\Sigma_i|^{1/2}} - tr(\Sigma^{-1}\mathbf{S}) \quad (10)$$

where $tr(\cdot)$ denotes the trace of matrix, Σ is a $CJ \times CJ$ diagonal covariance matrix whose diagonal blocks are Σ_i. $\Omega(\mathbf{z}(\lambda))$ encompasses all terms related to $\mathbf{z}(\lambda)$, i.e.

$$\Omega(\mathbf{z}(\lambda)) = \mathbf{z}^*(\lambda)\Phi^*\Sigma^{-1}\Xi - \frac{1}{2}\mathbf{z}^*(\lambda)\Phi^* N\Sigma^{-1}\Phi\mathbf{z}(\lambda) \quad (11)$$

actually the occupation rate N and the first order statistics Ξ depend on λ, and $\Xi = \begin{pmatrix} \Xi_1 \\ \cdots \\ \Xi_C \end{pmatrix}$ where $\Xi_i = \sum_{t=1}^{\kappa}(\mathbf{x}_t - \mathbf{m}_i)$; and $\mathbf{S}$ is the second order statistics. As a result, the posterior distribution of the vector $\mathbf{z}(\lambda)$ given the observed variable $\mathbf{x}$ can be approximated by

$$P(\mathbf{z}|\mathbf{X}, \lambda) \propto P(\mathbf{X}|\mathbf{z}, \lambda)P(\mathbf{z}) \propto \exp(-\frac{1}{2}(\beta - \mathbf{z})^* \zeta(\lambda)(\beta - \mathbf{z})) \quad (12)$$

where $\beta = \zeta^{-1}(\lambda)\Phi^*\Sigma^{-1}\Xi$, and $\zeta(\lambda) = \mathrm{I} + \Phi^*\Sigma^{-1}N\Phi$, and I denotes identity matrix. This equation means: $\mathbf{E}\{\mathbf{z}|\mathbf{X}\} = \beta$, and $\mathrm{Cov}\{\mathbf{z}|\mathbf{X}\} = \zeta^{-1}(\lambda)$, where $\mathbf{E}$ denotes the expectation operator. We have

$$\hat{\mathbf{m}} = \mathbf{E}[\mathbf{m}(\lambda)] = \bar{\mathbf{m}} + (\gamma + N)^{-1}\Xi(\lambda, \mathbf{m}) \quad (13)$$

Comparing with the conventional MAP, (13) shows that the relevance factor γ can be estimated by using Φ and Σ, i.e. $\gamma = \Phi^{-2}\Sigma$. Φ can be estimated by computing the expectation-maximization (EM) algorithm as follows: The M-step for Φ is given by

$$\Phi = \Xi\mathbf{E}\left[\mathbf{z}^*(\lambda)\right](N\mathbf{E}\left[\mathbf{z}(\lambda)\mathbf{z}^*(\lambda)\right])^{-1} \quad (14)$$

and the E-step is

$$\mathbf{E}\{\mathbf{z}(\lambda)\} = [\mathrm{I} + \Phi^*\Sigma^{-1}N\Phi]^{-1}\Phi^*\Sigma^{-1}\Xi \quad (15)$$

$$\mathbf{E}\{\mathbf{z}^*(\lambda)\mathbf{z}(\lambda)\} = [\mathrm{I} + \Phi^*\Sigma^{-1}N\Phi]^{-1} + \mathbf{E}\{\mathbf{z}(\lambda)\}^2 \quad (16)$$

3.2. An adaptive relevance factor for MAP

The idea of MAP estimation for GMM was presented in [3]. The primary purpose of the MAP is to estimate the probability density function of a certain group of the data given a prior distribution. It is reasonable that for insufficient data the reliability is low so the value of α in (5) is small and the estimated GMM should be close to the UBM. When the data becomes sufficient, the reliability of the sufficient statistics is high so the value of α in (5) is large, so that the estimated GMM should be displaced further from the UBM. This is reflected by equations (4) and (5). Thus, when applying MAP to derive GMM supervector, to assure the reliability of the estimated model, the GMM supervector should be close to the UBM supervector when the feature data is insufficient, and vice versa.

However, in language recognition, usually GMM supervector is purposely used to represent the language of the utterance. It is generated from a universal language which is represented by the UBM supervector. This requires the distance from the universal language to the particular language does not vary with the length of the utterance. In other words, the GMM supervector is required to stably represent the characteristics of the particular language regardless of length of the utterance spoken. In short, ideally, each utterance with the same language is expected to give the same GMM supervector regardless of the duration of the utterance. In this way, the supervectors can stably represent the language without being affected by the duration of an utterance. Therefore, we propose an adaptive relevance factor as follows

$$\begin{aligned}\breve{r}_i^{(\rho)} &= r_i^{(\rho)}\varphi(\kappa) \\ &= r_i^{(\rho)}\{\varphi(\kappa_0) + \frac{\varphi'(\kappa_0)}{1!}(\kappa-\kappa_0) + \frac{\varphi''(\kappa_0)}{2!}(\kappa-\kappa_0)^2 + \cdots\}\end{aligned} \tag{17}$$

where φ is infinitely differentiable in a neighborhood of κ_0 which can be approximated with the average size of the utterances. According to (6), when κ increases, the probabilistic count N_i increases. Take the expectation of the N_i, we have

$$\mathbf{E}(N_i) = \mathbf{E}(\sum_{t=1}^{\kappa} \frac{\omega_i f(\mathbf{x}_t|\mathbf{m}_i,\Sigma_i)}{\sum_{j=1}^{M}\omega_j f(\mathbf{x}_t|\mathbf{m}_j,\Sigma_j)}) \propto \kappa \tag{18}$$

where $\mathbf{E}$ is the expectation operator. If we chose $\varphi(\kappa) \approx \theta_0\kappa$ by ignoring the high order polynomial terms we can arrive at

$$\mathbf{E}(\breve{\alpha}_i) \propto \frac{\mathbf{E}(N_i)}{\mathbf{E}(N_i) + \theta_0\kappa\Phi_i^{-2}\Sigma_i} \longrightarrow constant\ vector \tag{19}$$

where θ_0 is a constant value which can be obtained from the known database. It means the expectation of α can be stable when we have the relevance factor $\breve{r}_i^{(m)}$ as follows

$$\breve{\gamma}_i \approx \theta_0\kappa\Phi_i^{-2}\Sigma_i \tag{20}$$

This ensures that the distance measure between the GMM supervector and UBM supervector is not seriously affected by the length of the adaptation utterance.

4. Hybrid Pair Language Recognition System

In this paper, we develop a hybrid pair language recognition system using Bhattacharyya-based kernel. The particular construction of the system is illustrated in Fig. 1.

We use two strategies for SVM modeling for pair language recognition. First, we use core-to-pair modeling. This is done by generating η target models for target languages followed by pair-language measure. In the target model phase, the score is computed to form score vector according to the target model sequence. Each dimension of the score vector indicates the log likelihood of a target language for the given test segment. The language pair score is obtained by forming the log likelihood ratio between pair of languages. To this end, the component classifiers are trained to model and discriminate one language from the others (i.e., one-vs-all), among $\eta = 24$ language classes. The Gaussian backend is trained for each component classifier (η Gaussians with tied covariance matrices, which lead to the so-called linear backend) using development dataset for training score collection [10]. The result from the above calibration and fusion step is the η-dimensional log-likelihood vector

$$\mathbf{s}(t) = [s_1(t), s_2(t), \cdots, s_\eta(t)]^T; \tag{21}$$

where each element $s_n(t)$ in the score vector indicates the log-likelihood of the nth language class given the t-th test segment. Language pair scores can therefore be obtained by forming the log-likelihood ratio between pair of languages, according to Bayes' rule, as follows

$$S_{pair}(L_i, L_j, t) = \log p(\mathbf{X}_t/L_i) - \log p(\mathbf{X}_t/L_j) = s_i(t) - s_j(t) \tag{22}$$

The second strategy is called as pair-modeling where the components classifiers are trained to model directly the language pair. For η = 24, the number of languages pairs are $\eta(\eta-1)/2$ = 276. The final score are obtained by adding the scores of the two sets (the first based on the one-vs-all and the second based one-vs-one modeling strategies as described above) with equal weights [1]

4.1. Bhattacharyya-based GMM-SVM kernel

While the conventional GMM-supervector is the stacked normalized mean vectors of the GMM, we extend the concept of the GMM-supervector with its element being a certain function, $\mathbf{g}$, of the mean, covariance and weight. The process for generating the generalized GMM-supervector is summarized in Fig. 2. The GMM-supervector is formed by concatenating the function vector $\mathbf{g}(\mathbf{m}_i, \Sigma_i, \omega_i)$ of the Gaussian components, i.e.,

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}(\mathbf{m}_1,\Sigma_1,\omega_1) \\ \mathbf{g}(\mathbf{m}_2,\Sigma_2,\omega_2) \\ \vdots \\ \mathbf{g}(\mathbf{m}_i,\Sigma_i,\omega_i) \\ \vdots \\ \mathbf{g}(\mathbf{m}_C,\Sigma_C,\omega_C) \end{bmatrix} \tag{23}$$

We refer to $\mathbf{g}(\mathbf{m}_i, \Sigma_i, \omega_i)$ as the ith subvector of the GMM-supervector. In this way, the generalized GMM-supervector maps a speech utterance to a high-dimensional vector.

In our previous work [8], we derived an Bhattacharyya-

[1] Equal weights are used since the scores are 'properly' calibrated log-likehood ratio, and both subsystems are equally important.

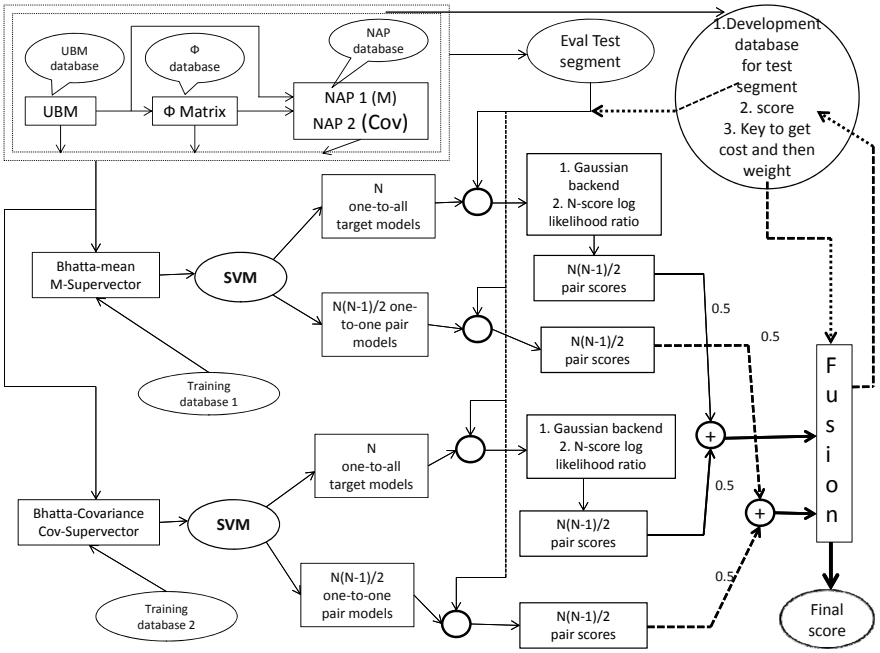

Figure 1: The hybrid pair language recognition system.

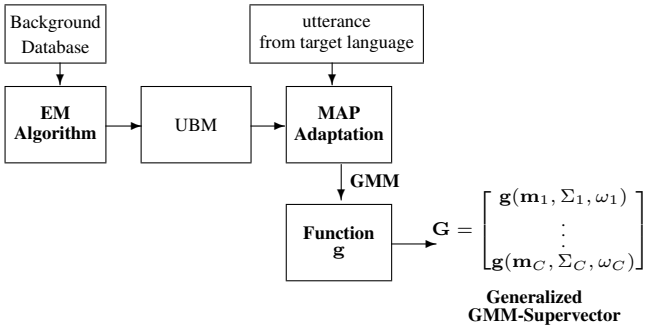

Figure 2: The process generating the generalized GMM-supervector from an utterance. $\mathbf{g}(\mathbf{m}_i, \Sigma_i, \omega_i)$ is the function vector of the mixture Gaussian parameters.

based distance between two GMMs as follows

$$\begin{aligned}
&\Psi_{\text{Bhatt}}(p_a||p_b) \\
&= \frac{1}{8}\sum_{i=1}^{C}\Big\{\Big[\Big(\frac{\Sigma_i^{(\text{a})}+\bar{\Sigma}_i}{2}\Big)^{-\frac{1}{2}}(\mathbf{m}_i^{(\text{a})}-\bar{\mathbf{m}}_i)\Big]^T \\
&\qquad \Big[\Big(\frac{\Sigma_i^{(\text{b})}+\bar{\Sigma}_i}{2}\Big)^{-\frac{1}{2}}(\mathbf{m}_i^{(\text{b})}-\bar{\mathbf{m}}_i)\Big]\Big\} \\
&+ \frac{1}{2}\sum_{i=1}^{M} tr\Big[\Big(\frac{\Sigma_i^{(\text{a})}+\bar{\Sigma}_i}{2}\Big)^{\frac{1}{2}}(\Sigma_i^{(\text{a})})^{-\frac{1}{2}}\Big(\frac{\Sigma_i^{(\text{b})}+\bar{\Sigma}_i}{2}\Big)^{\frac{1}{2}}(\Sigma_i^{(\text{b})})^{-\frac{1}{2}}\Big] \\
&+ \sum_{i=1}^{M}\ln\Big\{\frac{1}{\sqrt{\omega_i^{(\text{a})}\omega_i^{(\text{b})}}}\Big\} - \frac{M}{2}
\end{aligned} \tag{24}$$

Obviously, the distance is composed of two terms, i.e. the mean statistical dissimilarity and the covariance statistical dissimilarity. In order to avoid the unnecessary cross effect of the parameters, we consider that the mean statistical dissimilarity only carries the first-order of the adaptation data information with the mean vectors and the covariance statistical dissimilarity carries the second-order of new data information with the covariance matrices. Usually, the first term can be applied solely; we can assume that the covariance is not adapted and only exploit the mean information in the equation. By combining the two terms in (24), we arrive at the following kernel in practice

$$\begin{aligned}
&K_{\text{Bhatt}}(\mathbf{X}_a, \mathbf{X}_b) \\
&= \sum_{i=1}^{C}\Big\{\Big[\frac{1}{2}\Big(\bar{\Sigma}_i\Big)^{-\frac{1}{2}}(\mathbf{m}_i^{(\text{a})}-\bar{\mathbf{m}}_i)\Big]^T\Big[\frac{1}{2}\Big(\bar{\Sigma}_i\Big)^{-\frac{1}{2}}(\mathbf{m}_i^{(\text{b})}-\bar{\mathbf{m}}_i)\Big]\Big\} \\
&\quad + \sum_{i=1}^{C} tr\Big[\Big(\frac{\Sigma_i^{(\text{a})}+\bar{\Sigma}_i}{2}\Big)^{\frac{1}{2}}(\Sigma_i^{(\text{a})})^{-\frac{1}{2}}\Big(\frac{\Sigma_i^{(\text{b})}+\bar{\Sigma}_i}{2}\Big)^{\frac{1}{2}}(\Sigma_i^{(\text{b})})^{-\frac{1}{2}}\Big]
\end{aligned} \tag{25}$$

Considering the compensation effect benefited from the different database we develop the separated systems with the mean and covariance respective. We have the mean supervector that contains only mean variables.

$$\begin{aligned}
&K_{\text{Bhatt-mean}}(\mathbf{X}_a, \mathbf{X}_b) \\
&= \sum_{i=1}^{C}\Big\{\Big[\frac{1}{2}\Big(\bar{\Sigma}_i\Big)^{-\frac{1}{2}}(\mathbf{m}_i^{(\text{a})}-\bar{\mathbf{m}}_i)\Big]^T\Big[\frac{1}{2}\Big(\bar{\Sigma}_i\Big)^{-\frac{1}{2}}(\mathbf{m}_i^{(\text{b})}-\bar{\mathbf{m}}_i)\Big]\Big\}
\end{aligned} \tag{26}$$

And the covariance vector containing the covariance term only.

$$K_{\text{Bhatt-cov}}(\mathbf{X}_a, \mathbf{X}_b) = \sum_{i=1}^{C} tr\Big[\Big(\frac{\Sigma_i^{(a)} + \bar{\Sigma}_i}{2}\Big)^{\frac{1}{2}} (\Sigma_i^{(a)})^{-\frac{1}{2}} \Big(\frac{\Sigma_i^{(b)} + \bar{\Sigma}_i}{2}\Big)^{\frac{1}{2}} (\Sigma_i^{(b)})^{-\frac{1}{2}}\Big] \tag{27}$$

4.2. Support vector machine

Since each element of the GMM-supervector is Gaussian distributed, with the Bayesian minimum risk criterion the kernel scoring can be obtained by

$$\begin{aligned}\Gamma_{\text{cost}}(\mathbf{X}) &= \sum_{l=1}^{L} \alpha_l t_l K_{\text{GMM-Sup}}(\mathbf{X}_l, \mathbf{X}) + d \\ &= \Big(\sum_{l=1}^{L} \alpha_l t_l \mathbf{S}^{(\mathbf{X}_l)}\Big)^T \mathbf{S}^{(\mathbf{X})} + d \\ &= \mathbf{w}^T \mathbf{S}^{(\mathbf{X})} + d\end{aligned} \tag{28}$$

where t_l is the target value of +1 or -1 corresponding to the target class or non-target class, $\mathbf{X}_l$ is actually a sequence of feature vectors of utterance l. $\alpha_l > 0$ is the weight of the vector $\mathbf{X}_l$ so that $\sum_{l=1}^{L} \alpha_l t_l = 0$. d is a bias parameter independent of the observed sequence. $\mathbf{S}$ denotes supervector, here it represents the support vector.

Given a set of linearly separable two-class data, there are many possible solutions. An SVM is a binary linear classifier represented by a hyperplane separator. The separator is selected by maximizing the distance between the hyperplane and the closest training vectors. By introducing SVM, the $\mathbf{X}_l|_{l=1,\ldots,L}$ are selected from the training data and called the support vectors since they support the hyperplanes on both sides of the margin, and $\mathbf{w}$ is for the linear combination of the support vectors. The support vectors are obtained by a quadratic optimization [11] [12]. With the trained model represented by parameters $\mathbf{w}$ and d, the cost value in (28) is used as the score during recognition.

4.3. Nuisance attribute projection

In our pair language recognition system, NAP [13] [14] for channel compensation is used, it is applied to all GMM-supervectors. In language recognition, a commonly known phenomenon is that same language may be spoken with different microphones, different channel conditions and different environment backgrounds. NAP is used to reduce the session variability in the same language group by projecting it out based on eigen-decomposition. This is done by removing the subspace that causes the variability. It makes the GMM distribution distance more accurately reflect between-language distances. For a kernel $K_(p_a, p_b) = [\mathbf{S}^{(a)}]^T[\mathbf{S}^{(b)}]$, NAP constructs a new kernel by

$$\begin{aligned}K_{\text{NAP}}(p_a, p_b) &= [(\mathbf{I} - \mathbf{v}\mathbf{v}^T)(\mathbf{S}^{(a)})]^T[(\mathbf{I} - \mathbf{v}\mathbf{v}^T)\mathbf{S}^{(b)}] \\ &= [\mathbf{P}\mathbf{S}^{(a)}]^T[\mathbf{P}\mathbf{S}^{(b)}]\end{aligned} \tag{29}$$

where $\mathbf{v}$ is a matrix of eigenvectors estimated from within-class covariance matrix. The eigenvector matrix $\mathbf{v}$ is an orthonormal principal matrix with its rank set to a specified NAP-rank, corresponding to the NAP-rank largest eigenvalues. $\mathbf{P}$ is called as the projection matrix, and $\mathbf{I}$ denotes the identity matrix. In our experiment, the procedure used to estimate the NAP matrix is described as follow

1. Extract supervector for each language to group a set of supervectors;
2. Separate the set of supervector with the same language into several subgroups;
3. compute the mean of the supervectors for each subgroup;
4. Subtract the supervectors by the mean corresponding to each subgroup, so that language variability is removed;
5. Collect all the mean-removed supervectors to form a big matrix Ω where the intersession variability remains;
6. Do the eigen-decomposition of $\Omega\Omega^T$; so that $\mathbf{v}$ is obtained.

The projection matrix $\mathbf{P}$ is separately trained for each kernel by using the utterances selected from language recognition training database.

5. Performance Evaluation

Language recognition performance is measured for each target language pair. For each pair $(L_1 \leftrightarrow L_2)$, the miss probabilities for L_1 and for L_2 over all segments in either language will each be determined. In addition, these probabilities are to be combined into a single number that represents the cost performance of a language recognition system for distinguishing the two languages, according to an application motivated cost model.

We adopt NIST LRE 2011 [9] 30-second task to evaluate the performance of our pair language recognition system. The LRE 2011 evaluation emphasizes the language pair condition. It involves both conversational telephone speech (CTS) and broadcast narrow-band speech (BNBS), generally involving people telephoning into the broadcast studio. Multiple broadcast sources are included. The performance will be evaluated over a set of trials. Trials consist of a test segment along with a specified target language pair. The full set of trials consist of all combinations of an evaluation test segment and a target language pair. Thus if η is the number of target languages, each test segment is used for $\eta \times (\eta - 1)/2$ trials. An overall performance measure for each system will be computed as an average cost for those target language pairs presenting the greatest challenge to the system. For each duration, a systems overall performance measure will be based on the η target language pairs for which the minimum cost operating points for 30-second segments are greatest.

We use MFCC SDC with 56 and 80 dimensionality obtained with configurations of 7-1-3-7 and 10-2-3-7, respectively. These are then fused via a score fusion together with other component classifiers. For the 56-dim SDC, the UBM contains 512 mixtures. For the 80-dim SDC, the UBM consists of 1024 Gaussian mixtures. In addition to the mean vector, the covariance supervector is formed by concatenating the covariances, as detailed. NAP was applied on the supervectors for channel compensation. The NAP rank is selected to 70 for the GMM-supervector with GMM 1024 with 80 dimension SDC. These are then fused via score fusion together with other component classifiers.

In order to cover the variability of NIST-LRE evaluation data, our training data consist of two major partitions: Broadcast Narrow-Band Speech (BNBS) data and Conversational Telephone Speech (CTS). This dataset includes recordings from CallFriend, OHSU, previous NIST-LRE evaluations (96, 03, 05 and 07), OGI22, VOA (provided by NIST and additional download) and other BNBS data downloaded from different sources.

Table 1: The comparison of the pair language recognition systems using core-to-pair modeling in terms of EER and minimum detection cost for LRE 2011 30s task

LRE 2011, 30s, N-top average	*EER*	*min. Cost × 100*
Bhatt56:arf, core2pair	13.35 %	12.58
Bhatt80:rf=0.25, core2pair	12.92 %	12.02
Bhatt80:rf=8, core2pair	12.10 %	11.48
Bhatt80:rf=32, core2pair	13.80 %	13.65
Bhatt80:arf, core2pair	11.89 %	10.41

Table 2: The comparison of the pair language recognition systems using pair modeling in terms of EER and minimum detection cost for LRE 2011 30s task

LRE 2011, 30s, N-top average	*EER*	*min. Cost × 100*
Bhatt56:arf, pair	14.18 %	13.36
Bhatt80:rf=0.25, pair	13.83 %	13.14
Bhatt80:rf=8, pair	13.64 %	12.69
Bhatt80:rf=32, pair	14.65 %	14.08
Bhatt80:arf, pair	12.75 %	12.07

We use two different strategies to arrive at the final language pair scoring. In the experiment, we compare different Bhattacharyya-based GMM-SVM systems, and especial focus on the progress of each stage in the hybrid pair language recognition system. We name the Bhattacharyya-based system with GMM-1024 and 80 dimension of SDC feature as 'Bhatt80' or 'BhattCov80' for mean (26) or covariance (27) kernel. 'Bhattall80' denotes the combination of both mean and covariance kernels at score level. and the one with GMM-512 and 56 dimension of SDC as 'Bhatt56'; the system with adaptive relevance factor is denotes by 'arf'; and the fixed relevance factor with value k is denoted by 'rf=k' (for instance, when k=8, it is denoted by 'rf = 8'). We use 'core2pair' to denote core-to-pair modeling, and 'pair' denotes the pair modeling. The EERs and detection costs listed in tables 1 and 2 give a comparison between fixed relevance factor and adaptive relevance factor based on core-to-pair modeling and pair modeling respectively. It can be seen that the adaptive relevance factor brings better performance than fixed ones. Besides the different relevance factor comparison, comparing the first and last lines in the two tables shows that GMM 1024 with 80 dimension SDC gives more effectiveness than GMM 512 with 56 dimension SDC does. Obviously, the phenomenon that the higher dimension feature achieves a higher performance is simply due to more information introduced into the supervector. The above observation can be also viewed from Figs 3 and 4.

The results given in table 3 show the progressive situation in different stages of the hybrid system. Figs 5, 6 and 7 show the same observation in terms of top-η average minimum detection cost for the hybrid pair language recognition system. The improvement in each stage is apparent.

Table 3: The combination of mean and covariance results in score level as well as the final fusion of the combined core-to-pair modeling and the combined pair modeling in the hybrid pair language recognition system in terms of EER and minimum detection cost on LRE 2011 30s evaluation platform

LRE 2011, 30s, N-top average	*EER*	*min. Cost × 100*
Bhatt80:arf, core2pair	11.89 %	10.41
BhattCov80:arf, core2pair	20.36 %	19.00
Bhattall80, arf, core2pair	10.78 %	10.00
Bhatt80:arf, pair	12.75 %	12.07
BhattCov80:arf, pair	23.81 %	22.72
Bhattall80, arf, pair	12.30 %	11.43
Fusion: Bhattall80(core + pair)	10.08 %	9.02

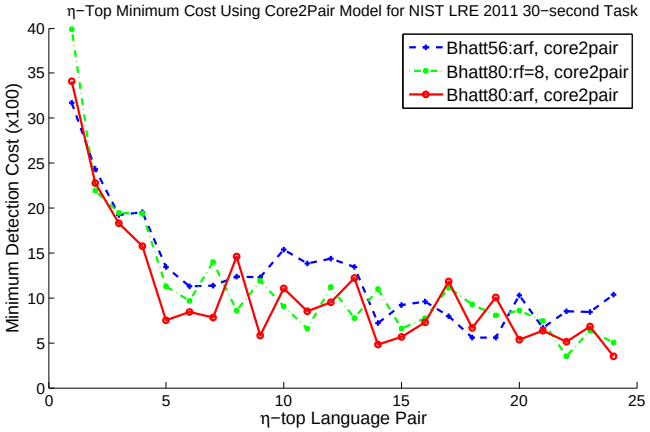

Figure 3: comparison the top-η detection cost between different Bhattacharyya systems with the core-to-pair modeling.

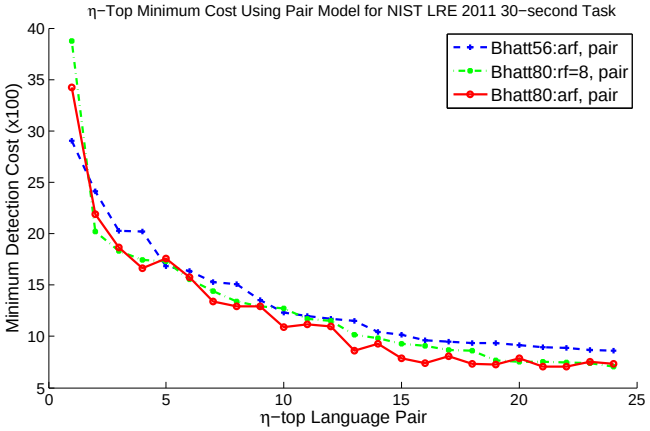

Figure 4: comparison the top-η detection cost between different Bhattacharyya systems with the pair modeling.

6. Summary

We have developed a hybrid Bhattacharyya-based GMM-SVM

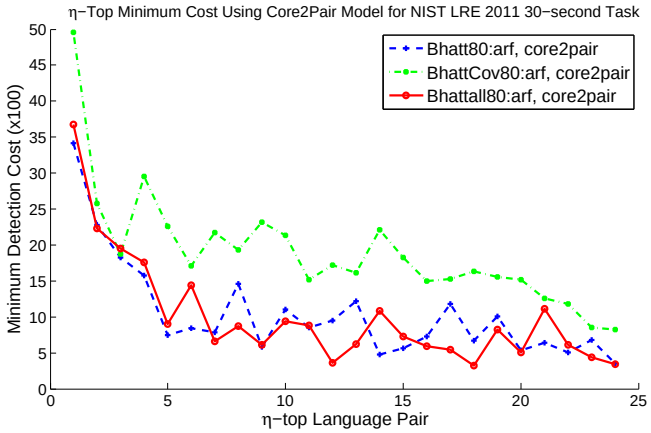

Figure 5: Complementary of the mean and covariance kernels with core-to-pair modeling.

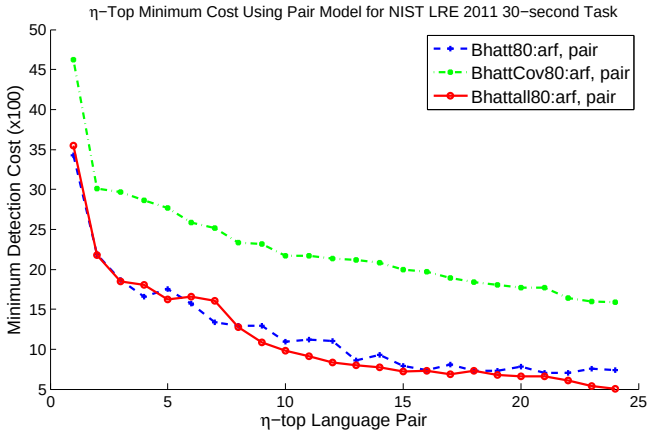

Figure 6: Complementary of the mean and covariance kernels with pair modeling.

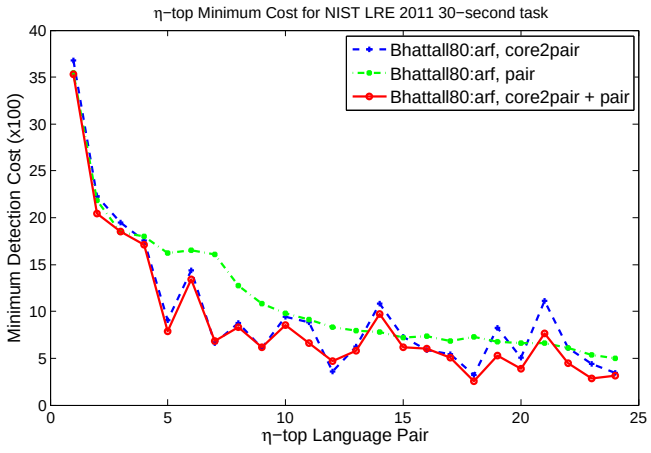

Figure 7: Fusion of the core-to-pair modeling and pair modeling systems.

system for pair language recognition. It can be viewed from the experiment that the higher dimension feature can surely reach a higher performance, since there are more information captured into the supervector. In GMM-SVM language recognition system, a GMM supervector is used to represent the language property of a speech segment (or utterance) and serves as an input vector to the SVM. This requires the elimination of the negative effect of the database variation in order to manifest the saliency of the language characteristics. We described the data-dependent relevance factor of MAP in supervector domain and introduced the adaptive relevance factor for GMM. We investigated the effectiveness of the adapted relevance factor as compared to the fixed relevance factor on the pair language recognition platform. Moreover, the improvement after the merge of mean supervector and covariance supervector is obvious. It is also observed that the core-to-pair modeling has a great complementarity with the pair modeling. In a word, it has been shown that the developed hybrid pair language recognition system gives very effective performance. The efficacy of the adaptive relevance factor as well as the hybrid pair language recognition system is shown by using the Bhattacharyya-based SVM kernel on the LRE 2011 30-second task.

7. References

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

[2] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," *Int. Conf. on Spoken Lang. Process.*, pp. 89-92, 2002.

[3] J. L. Gauvain and C-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, pp. 291-298, 1994.

[4] P. Kenny, "Joint Factor Analysis of Speaker and Session Variability : Theory and Algorithms," CRIM, Montreal, *Technical Report, CRIM-06/08-13*, 2005.

[5] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digit. Signal Process.*, vol. 10, pp. 19-41, 2000.

[6] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, "Language Recognition with Support Vector Machines," *Proc. Odyssey: The Speaker and Lang. Recog. Workshop* Toledo, pp. 41-44, 2004.

[7] C. H. You, H. Li, and K. A. Lee, "A GMM-supervector approach to language recognition with adaptive relevance factor," *18th Europ. Signal Process. Conf.*, EUSIPCO, pp. 1993-1997, Aalborg, Aug. 2010.

[8] C. H. You, K. A. Lee and H. Li, "GMM-SVM Kernel with a Bhattacharyya-Based Distance for Speaker Recognition," *IEEE Trans. Audio, Speech and Lang. Process.*, vol 18, no. 6, pp. 1300-1312, Aug. 2010.

[9] http://www.itl.nist.gov/iad/mig/tests/lre/2011/

[10] N. Brümmer, "FoCal Multi-class: Toolkit for Evaluation, Fusion and Calibration of Multi-class Recognition Scores," Available: http://niko.brummer.googlepages.com/focalmulticlass.

[11] C. C. Chang and C. J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, pp. 1-27, Apr. 2011.

[12] R. Collobert and S. Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143-160, 2001.

[13] A. Solomonoff, W. M. Campbell, and C. Quillen, "Channel compensation for SVM speaker recognition," *Proc. Odyssey04*, pp. 57-62, 2004.

[14] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," *Int. Conf. Acoust. Speech and Signal Process.*, 2005.