PAPER

Error Corrective Fusion of Classifier Scores for Spoken Language Recognition

Omid DEHZANGI[†a)], **Bin MA**[††b)], **Eng Siong CHNG**[†c)], *Nonmembers*, *and* **Haizhou LI**[†,††,†††d)], *Member*

SUMMARY This paper investigates a new method for fusion of scores generated by multiple classification sub-systems that help to further reduce the classification error rate in Spoken Language Recognition (SLR). In recent studies, a variety of effective classification algorithms have been developed for SLR. Hence, it has been a common practice in the National Institute of Standards and Technology (NIST) Language Recognition Evaluations (LREs) to fuse the results from several classification sub-systems to boost the performance of the SLR systems. In this work, we introduce a discriminative performance measure to optimize the performance of the fusion of 7 language classifiers developed as IIR's submission to the 2009 NIST LRE. We present an Error Corrective Fusion (ECF) method in which we iteratively learn the fusion weights to minimize error rate of the fusion system. Experiments conducted on the 2009 NIST LRE corpus demonstrate a significant improvement compared to individual sub-systems. Comparison study is also conducted to show the effectiveness of the ECF method.

key words: *Spoken Language Recognition, classifier fusion, error corrective training, error minimization*

1. Introduction

Spoken Language Recognition (SLR) is a classification problem to automatically identify the language of a spoken utterance [1]–[3]. In recent years, SLR has become an essential technology in many applications such as in multilingual spoken dialog systems, spoken language translation, automatic call routing and spoken document retrieval.

Typical SLR systems consist of two major modules: front-end feature extraction and back-end classifier. In current state-of-the-art SLR systems, three types of features are widely used. Acoustic features such as Mel-frequency Cepstral Coefficients (MFCCs), Shifted Delta Cepstral (SDC) features [4], high-dimensional feature vector using the Generalized Linear Discriminant Sequence (GLDS) kernel [5], phonotactic features such as language model scores of phone n-grams [2], phonotactic statistics of phone n-grams [3], and prosodic features which refer to long time acoustic structures such as in [6], [7]. Extracted features are fed to classifiers such as Gaussian Mixture Models (GMM) [8], Support Vector Machines (SVM) [9], [3], Hidden Markov Models (HMM) [10] to develop various SLR systems such as Parallel Phone Recognition followed by n-gram Language Model [2], Parallel Phone Recognition followed by Vector Space Model [3], phone Recognition with discriminative keyword selection using recursive feature elimination [11], using Latent Factor Analysis (LFA) [12]/Nuisance Attribute Projection (NAP) [13], using GMM supervectors followed by SVM [14], etc.

As a wide variety of effective systems have been developed in SLR, it has been a common practice to apply fusion systems in recent National Institute of Standards and Technology (NIST) Language Recognition Evaluations (LREs) [15]. In such systems, the results from individual classifiers utilizing different features and classifiers are fused to generate the final recognition results. The basic idea of fusion is to combine multiple decisions generated by different experts in an attempt to improve the performance of the overall system. The key issue to design a suitable and effective fusion scheme is to appropriately exploit all the available discriminative cues to generate an enhanced recognition result. Recently, successful SLR systems using fusion techniques have been introduced [16]–[18].

Information fusion can be carried out at three levels of abstraction closely connected with the flow of the classification process: data level fusion, feature level fusion, and classifier fusion [19]. This paper focuses on the study of the classifier fusion. The fusion of multiple classifiers operates as a mixture of experts to make collective decisions by exploiting information from each individual classifier. Figure 1 shows a block diagram of such a system. The input feature vectors to each different classifier are generated by different speech front-ends. If the classifier outputs offer complementary information, classifier fusion can improve the performance. A number of fusion techniques with considerable improvements over a single system have been proposed in speaker and spoken language recognition based on linear score weighting [20], GMM [18], SVM [21], ANN [22], etc. in which the optimized weighting coefficients are applied to the scores produced by individual classifiers.

In SLR, Detection Error Tradeoff (DET) curve is commonly employed to report the performance of SLR systems [23]. In this paper, we aim to improve the DET curve of the classifier fusion system by adjusting the fusion weights. An operating point on the DET curve is determined by a

Manuscript received March 23, 2011.
Manuscript revised July 22, 2011.

[†]The authors are with School of Computer Engineering, Nanyang Technological University, Singapore, 639798.
[††]The authors are with Institute for Infocomm Research, Singapore, 138632.
[†††]The author is with the Department of Computer Science and Statistics, University of Eastern Finland, FI-80101 Joensuu, Finland.
a) E-mail: dehzangi@pmail.ntu.edu.sg
b) E-mail: mabin@i2r.a-star.edu.sg
c) E-mail: aseschng@ntu.edu.sg
d) E-mail: hli@i2r.a-star.edu.sg
DOI: 10.1587/transinf.E94.D.2503

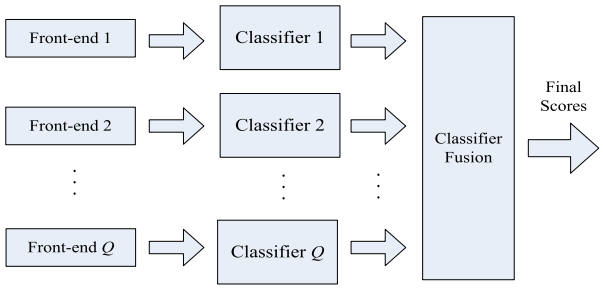


Fig. 1 Block diagram for a classifier fusion system.

decision threshold. The performance of such an operating point (e.g. false acceptance and false rejection error rates as well as a desired tradeoff between the two error types) is measured by the detection cost function [24]. The best operating point on a DET curve is the one that yields the lowest cost. We present an alternative fusion method for SLR in which an Error Corrective Fusion (ECF) algorithm is proposed to iteratively learn the weights assigned to the output scores of the classifiers so that the error rate of the classifier fusion is minimized. The proposed iterative algorithm learns the score weights one by one by finding the best operating point of each sub-system (classifier) for each specific target language (class). At each step assuming that the weights of all other scores are given and fixed, the resulting weight is optimal in the sense that it minimizes the detection error rate of the classifier fusion on the training data.

We will compare the proposed ECF method to the FoCal system [20], [25] which is currently the most popular method for fusion and calibration of detection scores for speaker and spoken language recognition. FoCal estimates the weighting coefficients of the output scores using a conjugate gradient descent algorithm to minimize a linear logistic regression function. On the other hand, ECF directly relates the performance of the system to the fusion parameters and determines the weights one at a time iteratively by solving a 2-class, one-dimensional problem to minimize the errors of the classifier fusion. FoCal employs a linear learning mechanism to optimize the objective function, while ECF makes use of a non-linear learning algorithm to correct the errors of the classifier fusion. We expect that the ECF learning mechanism leads to improve the performance of the SLR system by taking into account the inter-language discriminative information in the non-linear subspaces at the score level.

The organization of this paper is as follows. In Sect. 2, we introduce the spoken language recognition task and the employed sub-systems. In Sect. 3, the ECF framework is presented and the process of learning the fusion score weights is described. In Sect. 4, some other fusion strategies are introduced for comparison. In Sect. 5, the experimental results are presented. Finally, Sect. 6 concludes the paper.

2. Spoken Language Recognition

SLR technology has advanced tremendously in recent years, as evidenced by the results in the NIST LREs [15]. In this paper, we study the language recognition problem as formulated in the NIST evaluation campaign, in which, given a segment of speech and a language hypothesis, the task is to automatically decide whether the hypothesis correctly identifies the language spoken in the speech segment.

2.1 Task and Corpus

In the 2009 NIST LRE evaluation [24], 23 target languages and 16 non-target languages were involved in the 41793 test segments. Each test segment has a duration of either 30, 10, or 3 seconds. The test data are from either conversational telephone speech or Voice of America radio broadcasts.

We examine the IIR (Institute for Infocomm Research, Singapore) submission as the case study for the proposed classifier fusion strategy, and report the system performance using the Equal Error Rate (EER) measure, at which the system performs with the equal false acceptance rate and false rejection rate. In this study, the results are reported based on the core task where each of the test segments belongs to one of the 23 target languages.

2.2 Description of the Sub-Systems

A spoken language can be recognized using discriminative cues obtained from multiple sources. It is generally agreed upon that the integration of different discriminative cues can improve the performance of language recognition [16]–[18]. In the IIR's submission to the 2009 NIST LRE [26], 7 language classifiers were developed for the language recognition, as follows:

1. **Bhatt-SVM** is an implementation of the GMM-SVM language classifier derived based on the Bhattacharyya distance between GMMs [27].
2. **Extended Bhatt-SVM coupled with model pushing** is an extension of the Bhatt-SVM to include the covariance matrices as part of the supervector [28] and a pushback strategy [29].
3. **GLDS-SVM** extracts the feature vectors from an utterance that is expanded to a higher dimensional space using the Generalized Linear Discriminant Sequence (GLDS) kernel to form the final input to SVM classifiers [5].
4. **LM-GMM coupled with joint factor analysis** trains the target language GMMs with the large margin estimation where the multi-class separation margin is defined as the likelihood distance between true language model and the closest false language model [30]. The eigenchannels, which are trained with the joint factor analysis (JFA) approach [31], transform the feature vectors to be channel-independent.
5. **PW-SVM** uses the posterior weights [32] of a GMM to represent the input feature conveyed by a speech utterance, instead of using the mean vectors to represent the speech utterances as in Bhatt-SVM.
6. **PPRVSM** is a system with six parallel phone recognizers developed in IIR as the front-end and using a vector

space modeling method to model phonotactic information [33].

7. **PPRVSM-TALM** is another PPRVSM system and is different only in the frontend phone recognizers used. Three phone tokenizers are derived from Hungarian phone recognizer using Target-Aware Language Models (TALM) [34].

The above classifiers make use of either phonotactic features which are extracted to represent phonetic constraints in a language, or acoustic features which represent the spectral properties of speech spectrum to produce output scores. The details of the individual classifiers are not the focus in this paper; therefore, sub-system 1 to 7 will be used instead of the names of the classifiers hereafter.

3. The Error Corrective Fusion (ECF) System

Given an SLR problem with M target languages involved and Q individual language classifiers for the language recognition task trained using the labeled speech segments. The i-th classifier maps the speech segment $\mathbf{x}$ to a score vector $S_i(\mathbf{x}) = \{S_{i,j}(\mathbf{x}) | j = 1, 2, \ldots, M\}$, in which each element is the relative log-likelihood associated with one of the target languages. Figure 2 shows the structure of the classifier fusion. The fusion output score for the target language j, $Score_j(\mathbf{x})$, is the linear weighted sum of all the output scores of the sub-systems,

$$Score_j(\mathbf{x}) = \left\{ \sum_{i=1}^{Q} w_{i,j}.S_{i,j}(\mathbf{x}) \right\}, j = 1, 2, \ldots, M \qquad (1)$$

where $S_{i,j}(\mathbf{x})$ is the log-likelihood score of the test segment $\mathbf{x}$ associated with the classifier i and target language j, and $w_{i,j}$ is the weight corresponding to $S_{i,j}(\mathbf{x})$. However unlike conventional linear score weighting techniques, the ECF weighting coefficients are language-dependant (e.g., the weighting coefficients vary for different languages and sub-systems) to employ inter-language discriminative information. In this way, it may reflect how one sub-system contributes to each particular language. There are a total of $Q \times M$ weighting coefficients to be learnt for this LRE task. A normalization process is also applied on the scores among M target languages. In the following section, we describe the proposed ECF approach for learning the fusion weights by a discriminative performance measure.

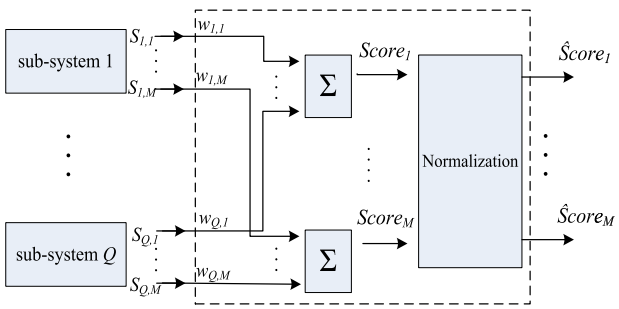


Fig. 2 Architecture of the ECF score fusion system.

3.1 The Discriminative Performance Measure

In this section, we propose a discriminative measure minimizing errors of a 2-class problem (separating the target language from its competing languages) to reduce the total EER. Figure 3 shows the confusion matrix for a 2-class classification problem (with positive '*pos*' and negative '*neg*' class labels), given a set of P positive and N negative labeled speech segments. The four different categories of decision results in Fig. 3 are as follows: *TP* (True Positives) indicating samples correctly labeled as positive; *FP* (False Positives) indicating negative samples incorrectly labeled as positive; *TN* (True Negatives) indicating samples correctly labeled as negatives; and *FN* (False Negatives) as the positive samples incorrectly labeled as negatives. The performance of a classifier can be extracted from the confusion matrix by defining a performance measure. For instance, the error rate of the classifier is defined as:

$$error\_rate = \frac{FP + FN}{TP + FP + TN + FN} \qquad (2)$$

Generally, a statistical classifier generates likelihood of positive class, $p(\mathbf{x}|'pos')$, and negative class, $p(\mathbf{x}|'neg')$, for an input speech segment $\mathbf{x}$ denoting the estimated probabilities that $\mathbf{x}$ belongs to positive and negative classes, respectively. We can define a measure called $negativity(\mathbf{x})$ as follows:

$$negativity(\mathbf{x}) = \frac{p(\mathbf{x}|'neg')}{p(\mathbf{x}|'pos')} \qquad (3)$$

The measure $negativity(\mathbf{x})$ demonstrates the degree to which $\mathbf{x}$ is believed to be of the negative class. The speech segment $\mathbf{x}$ is classified as negative if its $negativity(\mathbf{x})$ is greater than a specified threshold and positive otherwise. Since the likelihoods provided by the classifiers are not perfect due to deficient parameterization of the observations and insufficient data, the desired threshold needs to be learnt by optimizing a discriminative performance measure on available data. For instance, the *error_rate* in Eq. (2) corresponding to each specified threshold can be calculated and minimized.

We assume that a training set $\{\mathbf{x}_t | t = 1, 2, \ldots, n\}$ consisting of n labeled speech segments from M different languages is available. Using the training data, one is able

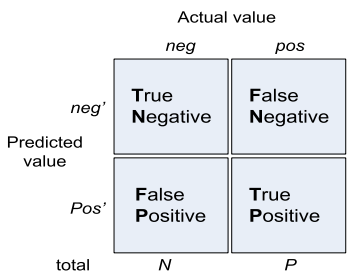


Fig. 3 Confusion matrix for 2-class problems.

to find the threshold to make the best decision for a classifier by varying the threshold from 0 to ∞. An efficient algorithm for calculating the best threshold in such a way has been proposed in [35] where the input speech segments are ranked in an ascending order of their *negativity*(.) measure $negativity(\mathbf{x}_1), \ldots, negativity(\mathbf{x}_{P+N})$. Considering any threshold between $negativity(\mathbf{x}_t)$ and $negativity(\mathbf{x}_{t+1})$, the first K segments will be classified as positive and the remaining P+N-K segments as negative. In this way, maximum of P+N+1 different thresholds need to be examined to find the best threshold. The first threshold classifies everything as negative and the last threshold classifies everything as positive. The rest of the thresholds are chosen in the middle of two successive measures. The threshold, *min_thresh*, on the *negativity*(.) measure is found such that it minimizes the error rate of the classifier, Eq. (2). The speech segment $\mathbf{x}_t$ is classified as positive if $negativity(\mathbf{x}_t) < min\_thresh$. That is, $\mathbf{x}_t$ is classified as $'pos'$ if,

$$\frac{p(\mathbf{x}|'neg')}{p(\mathbf{x}|'pos')} < min\_thresh \rightarrow p(\mathbf{x}|'neg') < min\_thresh \times p(\mathbf{x}|'pos') \tag{4}$$

In this way, *min_thresh* can be used as the weight of the positive class, $'pos'$.

3.2 Learning the Fusion Weights

Figure 4 shows the process of learning the fusion weights. As can be seen, the ECF learning algorithm uses feedbacks from final fusion scores to tune the fusion weights. The discriminative measure to find the best operating point in 2-class problems introduced in Sect. 3.1 will be used as an ingredient in the ECF learning algorithm to learn the fusion weights of the general M-class problem. The language recognition results are reported as the weighted average over multiple language detector sub-systems. The score distribution of each sub-system may be different due to employing different classifiers and speech front-ends. This makes the scores less comparable across different sub-systems. Hence, the score normalization is a necessary step leading to consistency over scores. In this way, the $Score_\varphi(\mathbf{x})$ from Eq. (1) is converted to log-likelihood ratio (LLR) $\hat{S}core_\varphi(\mathbf{x})$ as follows,

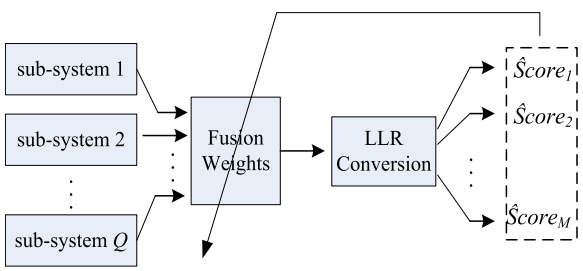


Fig. 4 The training paradigm for learning the fusion weights.

$$\hat{S}core_\varphi(\mathbf{x}) = Score_\varphi(\mathbf{x}) - log\left\langle \frac{1}{M-1} \sum_{j=1, j\neq\varphi}^{M} exp\left(Score_j(\mathbf{x})\right)\right\rangle \tag{5}$$

The above conversion is considered as the score normalization step as presented in [8]. The speech segment $\mathbf{x}$ is classified as the target language φ if,

$$\hat{S}core_\varphi(\mathbf{x}) > \theta_\varphi \tag{6}$$

where θ_φ is the decision threshold for the target language φ to be learnt on a development data set. The weight $w_{i,\varphi}$ corresponding to the classifier output score $S_{i,\varphi}(\mathbf{x})$ can be considered as the degree to which $S_{i,\varphi}(\mathbf{x})$ contributes in the classification decision. From (6), we can directly relate the fusion weight $w_{i,\varphi}$ to the final classification decision as follows,

$$w_{i,\varphi} > \frac{\theta_\varphi + log\left\langle \frac{1}{M-1} \sum_{j=1,j\neq\varphi}^{M} exp\left(-Score_j(\mathbf{x})\right)\right\rangle - \sum_{k=1,k\neq i}^{Q} w_{k,\varphi}.S_{k,\varphi}(\mathbf{x})}{S_{i,\varphi}(\mathbf{x})} \tag{7}$$

That is, $w_{i,\varphi}$ has to be greater than the right side of the inequality so that the input segment $\mathbf{x}$ is classified as the target language φ.

In the following, we present an algorithm that attempts to minimize the classification error rate of the classifier fusion by adjusting the fusion weights in the interval [0,∞) using the training speech segments. The weights assigned to each output score $S_{i,j}(.)$ is set to one (i.e. $w_{i,j} \leftarrow 1$ for $i = 1, \ldots, Q$, and $j = 1, \ldots, M$) as an initial solution to the problem. Then, the error rate of the classifier fusion is successively reduced by finding a better solution than the current one by learning the best fusion weights (resulting in minimum classification error rate) one at a time. In the following, we present a procedure that determines the optimal weight of a classifier score assuming that the weights of all others are given and fixed (i.e. locally optimum weight). Note that, by optimizing a single fusion weight, a better solution to the problem is presented. To find the weight $w_{i,\varphi}$ corresponding to output score of clssifier i for target language φ, the problem is considered as a 2-class problem where class φ is the positive class and class $\bar{\varphi}$ is the negative one. The steps are as follows:

1. The $w_{i,\varphi}$ is set to zero (i.e. $S_{i,\varphi}(.)$ does not contribute in classification decision).
2. The speech segments belonging to language φ that are classified correctly (*TP*) with current values of the classifier score weights are removed from the training set. Note that contribution of $w_{i,\varphi}$ can only help classify them correctly ($w_{i,\varphi} \in [0, \infty)$). We considered $w_{i,\varphi} = 0$ while generating the confusion matrix. Therefore, *TP* segments are classified correctly with no help from $w_{i,\varphi}$. Including them in estimating $w_{i,\varphi}$ not only is unnecessary but may even lead the learning algorithm to a sub-optimal solution.

3. The speech segments of $\bar{\varphi}$ that are misclassified (*FP*) are removed from the training set. Note that these patterns will be misclassified regardless of the value of $w_{i,\varphi}$ and contribution of $w_{i,\varphi}$ can only make it worse. Hence, they are not included in estimating $w_{i,\varphi}$.
4. The m speech segments left in the training set $\{x_t | t = 1, \ldots, m\}$ (i.e. *TN* and *FN*) are essential in estimation of $w_{i,\varphi}$. We need to estimate $w_{i,\varphi}$ so that error rate is minimized over $\{x_t | t = 1, \ldots, m\}$. From Eq. (7), the measure $negativity_{i,\varphi}(.)$ is calculated for every segment in $\{x_t | t = 1, \ldots, m\}$ as follows:

$$negativity_{i,\varphi}(\mathbf{x}_t) = \frac{\left[\theta_\varphi + log\left\langle \frac{1}{M-1} \sum_{j=1, j\neq\varphi}^{M} exp\left(-Score_j(\mathbf{x})\right)\right\rangle - \sum_{k=1, k\neq i}^{Q} w_{k,\varphi}.S_{k,\varphi}(\mathbf{x})\right]}{S_{i,\varphi}(\mathbf{x})} \quad (8)$$

where, $negativity_{i,\varphi}(\mathbf{x}_t)$ is the amount of $w_{i,\varphi}$ necessary for $\mathbf{x}_t$ to be classified as the target language φ.

5. The training segments are ranked in an ascending order of their $negativity_{i,\varphi}(.)$ measure. A threshold is defined and initialized by zero. Then, assuming that $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are two successive segments in the list, a threshold is computed as,

$$thresh = [negativity_{i,\varphi}(\mathbf{x}_t) + negativity_{i,\varphi}(\mathbf{x}_{t+1})]/2 \quad (9)$$

We then move the threshold from the lowest score to the highest. For each of the thresholds, we measure the associated error rate of the classifier fusion. The value of the *min_thresh* (i.e. leading to the minimum error rate) is used as the optimal score weight $w_{i,\varphi}$ assuming that all other score weights are fixed.

The pseudocode shown in Fig. 5 summarizes the process of learning the fusion weight $w_{i,\varphi}$. The algorithm receives a set of training speech segments $\{x_t | t = 1, \ldots, n\}$ and results *min_thresh* as the new value for $w_{i,\varphi}$.

The search for the locally optimum combination of weights is conducted by optimizing the score weights one at a time and learning stops if no improvement to the current performance can be made. Note that the algorithm is dependant to the order of the score weights optimized during the training process. To avoid the error rate of the classifier fusion to be skewed, the order of weight optimization is fixed in the way that every consecutive fusion weights to be estimated are corresponding to different target languages. The ECF method determines the output score weights of the subsystems attempting to better discriminate between the segments of language φ and those of the rest by finding the best operating point of the classifier fusion (including as many *FN* segments and as few *TN* segments as possible). By doing so, we locally optimize the tradeoff between false acceptance and false rejection error rates. In our experiments, we show that error rate on the training data never increases during the optimization of the fusion weights by the ECF learning mechanism and will converge to a local minimum.

```
1  Input: the set of training speech segments, { x_t | t=1, ...,n }
2  w_iφ ← 0            // i.e. S_iφ does not contribute in classification decision
3  remove TP speech segments from the training set {x_1 ... x_n}
4  remove FP speech segments from the training set {x_1 ... x_n}
5  calculate the negativity_iφ(.) measure in Eq. (8) for the remaining speech segments {x_1 .. x_m}
6  rank {x_1 .. x_m} in ascending order of their negativity_iφ(.) measure
7  thresh ← 0          // classifying every segment in {x_1 ... x_m} as class φ̄
8  minimum ← Error_Rate ({x_1 ... x_m}, thresh)   // calculates the error rate corresponding
   to the specified threshold ( x_t is classified as positive iff negativity_iφ(x_t) < thresh )
9  min_thresh ← thresh
10 for t=1 to m-1  where  negativity_iφ(x_t) ≠ negativity_iφ(x_t+1)
11     thresh ← [negativity_iφ(x_t) + negativity_iφ(x_t+1)] / 2   // segments having negativity_iφ(.)
       < thresh are classified as class φ
12     current ← Error_Rate ({x_1 ... x_m}, thresh)
13     if current < minimum then
14         minimum ← current
15         min_thresh ← thresh
16     end if
17 end for
18 thresh ← negativity_iφ (x_m) + ε   // classifying every segment in {x_1 ... x_m} as class φ (ε is a
   positive number)
19 current ← Error_Rate ({x_1 ... x_m}, thresh)
20 if current < minimum then
21     minimum ← current
22     min_thresh ← thresh
23 end if
24 w_iφ ← min_thresh
```

Fig. 5 The algorithm to learn the fusion weight $w_{i,\varphi}$.

4. Comparable Fusion Strategies

In this section, we describe some other fusion methods for comparison. For an input speech segment $\mathbf{x}$, fusion score of each target language j is a linear combination of the scores $\{S_{i,j}(\mathbf{x}) | i = 1, \ldots, Q\}$ each of which is the relative log-likelihood associated with the target language j. Comparative methods in this paper range from simple non-trainable combiners to methods that require sophisticated training procedures.

- **Max Rule** results in the maximal predicted probability of success.

$$Score_j^{\max}(\mathbf{x}) = \max_{i=1}^{Q} S_{i,j}(\mathbf{x}) \quad (10)$$

- **Min Rule** yields the minimal predicted probability of success.

$$Score_j^{\min}(\mathbf{x}) = \min_{i=1}^{Q} S_{i,j}(\mathbf{x}) \quad (11)$$

- **Simple Sum** whereby the output scores of every individual classifier are summed up and the label that receives the highest score is the output of the fusion system.

$$Score_j^{\text{sum}}(\mathbf{x}) = \sum_{i=1}^{Q} S_{i,j}(\mathbf{x}) \quad (12)$$

- **Local Accuracy-based Weighting (LAW)** is originally proposed in [36] in which the local accuracy of

each individual sub-system is estimated given the input features using K-Nearest Neighbor (KNN) classifier. The final output is solely given by the most reliable classifier of the fusion system. A modified version of LAW was also presented [37] in which the output is a weighted mean of reliability estimate of each subsystem which is the fraction of correctly classified samples in the local region surrounding the test segment reported by KNN classifier. The output of the fusion system is then defined as,

$$Score_j^{\mathrm{LAW}}(\mathbf{x}) = \frac{\sum_{i=1}^{Q} \lambda_i . S_{i,j}(\mathbf{x})}{\sum_{i=1}^{Q} \lambda_i} \tag{13}$$

where λ_i is the reliability estimate of the sub-system i.

- **Support Vector Machine (SVM)** has shown to be effective in separating input vectors in 2-class problems [38], in which SVM effectively projects the vector $\mathbf{x}$ into a scalar value $f(\mathbf{x})$,

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i . \mathbf{x}) + d \tag{14}$$

where the vectors $\mathbf{x}_i$ are support vectors, $y_i = \{-1, 1\}$ are the correct outputs, N is the number of support vectors, $(.)$ is the dot product function, α_i are adjustable weights and d is a bias. Learning is posed as an optimization problem with the goal of maximizing the margin (i.e., the distance between the separating hyperplane and the nearest training vectors). As a combiner method, we concatenate the output scores of all the sub-systems to form the input super vector to SVM. Then, we assign one SVM for each of the M languages (e.g. One-Vs-Rest scheme) and train accordingly to get the final output scores,

$$Score_j^{\mathrm{SVM}}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i (\sup(\mathbf{x}_i) . \sup(\mathbf{x})) + d \tag{15}$$

where $\sup(\mathbf{x})$ is the super vector resulted by concatenating the output scores of all the sub-systems for input segment $\mathbf{x}$.

- **FoCal** is an approach for fusion and calibration of detection scores which has been successfully used for speaker and spoken language recognition [20], [25]. For each segment $\mathbf{x}$, scores of the sub-system i that are assumed to be the relative log-likelihood associated with the target languages form a score vector $S_i(\mathbf{x}) = \{S_{i,j}(\mathbf{x}) | j = 1, \ldots, M\}$. The calibration scheme aims to transform detection scores to LLRs. A calibration transformation function is then defined as follows:

$$Score_j^{\mathrm{FoCal}}(\mathbf{x}) = F(S_i(\mathbf{x}), \theta) = \alpha S_i(\mathbf{x}) + \beta \tag{16}$$

where α is a scalar, β is an M-dimensional vector and $\theta = (\alpha, \beta)$ constitute a set of transformation parameters, which are estimated by using a conjugate gradient descent algorithm to minimize a linear logistic regression function [18].

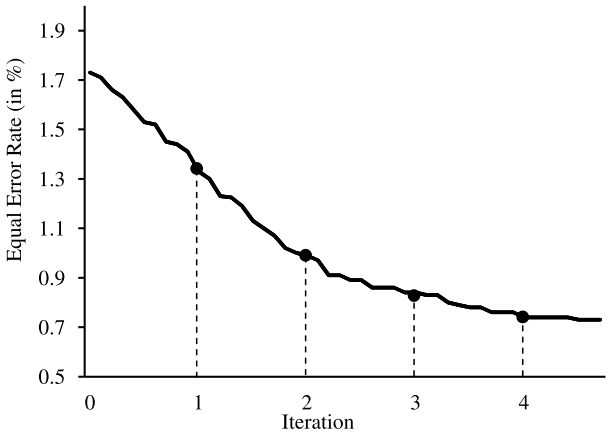
5. Experimental Results

We conducted the experiments on the 2009 NIST LRE evaluation data [24] including three tasks for the 30-, 10- and 3-second evaluation data sets. For each task, we have the training set, development set, and evaluation set. The training and the development sets which were extracted from the speech corpus provided by NIST, are used for training and parameter tuning of the system, and the evaluation set which is the test set of the 2009 NIST LRE, is only used to evaluate the performance of the system. The subsequent results are all based on the evaluation set which is not seen during training. We have used IIR's 7 language classifiers submission to the 2009 NIST LRE introduced in Sect. 2. There are $M = 23$ target languages in the LRE task. The output classifier scores are first scaled into the interval [0,1] to eliminate negative scores. In the following sections, we assess different aspects of the ECF learning process.

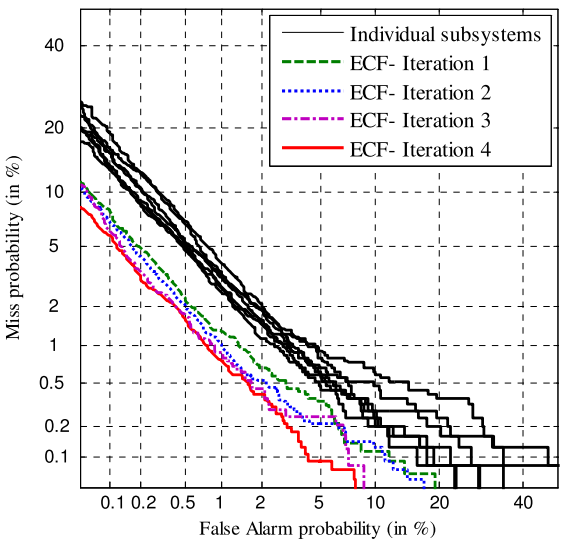
5.1 Investigating the Learning Process and Convergence of the ECF Method

In our first experiment, we investigated the convergence of the ECF learning mechanism during the training phase. We employed the ECF method to learn the fusion weights on the training set of 30-second speech segments and recorded the EER on the same data as the training progressed (close-test). Figure 6 (a) depicts the EER of the fusion system during the learning process for 4 training iterations. One training iteration denotes having all the $Q \times M$ fusion weights adjusted by the learning mechanism for one round. We chose 4 training iterations because it led to the best performance of the classifier fusion on the development set. Figure 6 (a) shows that the error rate decreases as we estimated the fusion weights one after another within each iteration. It can also be observed that the error curve is non-increasing as the learning continues and that the error curve becomes almost flat and converges after several training iterations. We also investigated the DET curve of the classifier fusion on the 30-second training set. Figure 6 (b) illustrates the DET curves of the classifier fusion by 4 iterations of the ECF method compared to the individual classifiers. It can be seen that the DET curves improves after each iteration of the ECF weight adjustments.

In another investigation, we verify the robustness of the ECF method to different initial weight values. In this section, we randomly initialize the fusion weights and then train the ECF system using the set of 10-second training speech segments for 4 iterations (chosen by experiment on the development set). We repeat the random initialization and training process for 30 times and draw the corresponding DET curves. Figure 7 shows the resulted DET curves of the trained ECF system with 30 different initial weight values. As illustrated in Fig. 7, there is no significant difference between the resulting DET curves which shows that the learning mechanism is not sensitive to the weight initial-



(a)



(b)

Fig. 6 (a) EER of the classifier fusion on the 30-second training data set during the ECF training process. (b) DET curve of the classifier fusion after each iteration of the ECF method. A training iteration denotes having all the fusion weights adjusted by the learning mechanism for one round.

ization.

5.2 The ECF Method Compared to the Individual Sub-Systems

In this section, we investigated the performance of the ECF method for score fusion compared to the individual sub-systems. Table 1 summarizes the averaged EERs achieved by the individual sub-systems and the ECF method for the case of training the ECF method on the 3-, 10-, and 30-second training sets and testing on the respective evaluation sets (open-test) for 4 iterations. The results in Table 1 suggest that the ECF method was successful to reduce the averaged EER of the fusion system significantly compared to the best individual sub-system in all three tasks. Table 1 also reports that in most of the cases, after each iteration of ECF training, the classifier fusion shows an improved results on

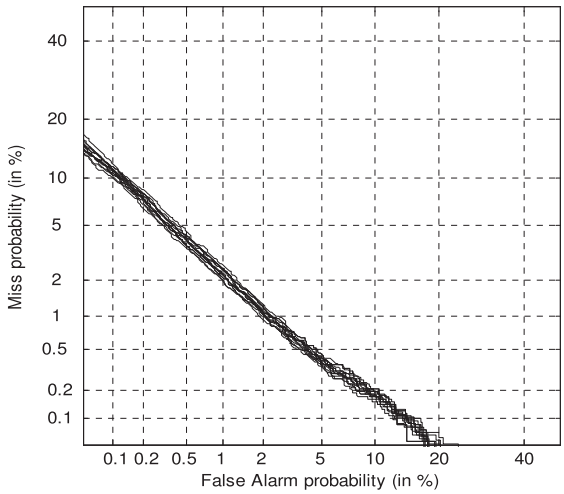


Fig. 7 DET curves corresponding to the trained ECF system with 30 different initial weight values.

Table 1 Average EER of 7 individual sub-systems and the ECF method.

System	30 sec	10 sec	3 sec
	Avg EER	Avg EER	Avg EER
Sub-system 1	3.23%	8.15%	20.35%
Sub-system 2	3.33%	8.11%	19.74%
Sub-system 3	4.39%	10.56%	24.20%
Sub-system 4	3.66%	6.60%	16.11%
Sub-system 5	3.61%	9.23%	22.94%
Sub-system 6	3.75%	7.84%	19.26%
Sub-system 7	3.94%	9.54%	23.99%
Best individual	3.23%	6.60%	16.11%
ECF-Iteration 1	2.47%	4.84%	13.66%
ECF-Iteration 2	1.94%	4.04%	12.14%
ECF-Iteration 3	1.83%	4.13%	11.73%
ECF-Iteration 4	1.72%	3.83%	11.62%

the evaluation set.

5.3 The ECF Method vs. the Comparative Fusion Systems

There are a variety of methods to fuse multiple classifiers. In this section, we compare the ECF method with some other fusion methods described in Sect. 4. Table 2 reports the results achieved by the ECF method with 6 other comparative fusion methods. The results in Table 2 suggest that trainable methods (LAW, SVM, FoCal, and ECF) produce substantial improvements to the best individual sub-system unlike the non-trainable strategies (Max rule, MIN rule, and Simple Sum) that do not seem to be helpful. As it is reported in Table 2, the ECF method provides the best averaged EER results among the comparative fusion strategies. It also outperforms the FoCal system moderately and the improvement is consistent over all the three tasks.

From Table 2, we can see that FoCal technique also offers a competitive result. It would be interesting to examine the difference between FoCal and ECF. We note that FoCal employs a linear learning mechanism to parameterize and optimize the objective function, while ECF method directly relates the recognition error through a non-linear learning

Table 2 Comparison of the averaged EER via ECF method with some other fusion techniques.

System	30 sec	10 sec	3 sec
	Avg EER	Avg EER	Avg EER
Best individual	3.23%	6.60%	16.11%
Max Rule	3.32%	6.57%	16.26%
Min Rule	3.19%	6.51%	15.37%
Simple Sum	3.13%	6.75%	17.07%
SVM	2.17%	5.06%	14.15%
LAW	2.29%	5.17%	13.83%
FoCal	1.81%	4.09%	12.27%
ECF method	**1.72%**	**3.83%**	**11.62%**

Table 3 SLR results of FoCal vs. ECF fusion with varying number of sub-systems.

System	30 sec	10 sec	3 sec
	Avg EER(%)	Avg EER (%)	Avg EER (%)
$Q = 2$			
FoCal	2.15 (1, 6)	5.11 (1, 6)	13.56 (4, 6)
ECF	2.06 (1, 6)	4.87 (1, 6)	12.97 (1, 6)
$Q = 3$			
FoCal	1.89 (1, 5, 6)	4.32 (2, 6, 7)	12.56 (1, 4, 6)
ECF	1.84 (1, 4, 6)	4.03 (1, 6, 7)	12.25 (1, 4, 6)
$Q = 4$			
FoCal	1.82 (1, 3, 5, 6)	4.03 (1, 4, 6, 7)	12.34 (1, 4, 6, 7)
ECF	1.78 (2, 3, 5, 6)	3.87 (2, 4, 6, 7)	11.86 (2, 4, 6, 7)
$Q = 5$			
FoCal	1.80 (2, 3, 5, 6, 7)	4.04 (1, 3, 4, 6, 7)	12.27 (2, 4, 5, 6, 7)
ECF	1.75 (1, 4, 5, 6, 7)	3.84 (1, 4, 5, 6, 7)	11.78 (1, 2, 5, 6, 7)
$Q = 6$			
FoCal	1.80 (1, 2, 3, 5, 6, 7)	4.05 (1, 2, 3, 4, 6, 7)	12.28 (1, 3, 4, 5, 6, 7)
ECF	1.73 (1, 2, 3, 5, 6, 7)	3.82 (1, 2, 3, 4, 6, 7)	11.67 (1, 2, 3, 4, 6, 7)

algorithm. This suggests that ECF allows for more flexible fitting when fusing multiple classifiers, each of which has its own dynamics. It is observed that ECF indeed leads to a better performance.

5.4 The ECF Method vs. FoCal with Varying Mixture of Sub-Systems

In this experiment, the ECF method and the FoCal system are both applied on varying number of sub-systems in the fusion. Table 3 illustrates the results achieved using only a subset of subsystems in the fusion. Q is the number of sub-systems contributed in the fusion (e.g. using only Q out of the 7 subsystems) and the numbers in the parentheses are the id number of the sub-systems with the best performance. The results show that as the number of sub-systems increase, the EER decreases or remains the same which is consistent with the information theory. However, the results show that the fusion of 4 sub-systems exploits most of the discriminative cues that the combination of all the sub-systems was capable of providing. Further increasing the number of subsystems in the fusion will only lead to minor improvement to the final results. This observation indicates that the information provided by different sub-systems is not entirely uncorrelated. Therefore by choosing the right mixture of subsystems in the fusion system, number of parameters of the fusion system required to achieve the optimal performance can be reduced. Table 3 also demonstrates that the ECF system consistently outperforms the Focal system over varying number of sub-systems in the fusion and different tasks.

6. Discussion

It has been a common practice to apply fusion methods utilizing different features and classifiers to generate the improved results in speaker and spoken language recognition. A variety of fusion techniques have been proposed in which the optimized weighting coefficients are applied to the scores produced by individual classifiers. As an example, FoCal system [20], [25] is currently the most popular method for fusion and calibration of detection scores in LRE.

In this paper, we proposed a novel error corrective fusion (ECF) system as an alternative fusion method for SLR in which an iterative learning algorithm was introduced to estimate the classifier fusion weights. We aimed to improve the ROC curve corresponding to the classifier fusion by tuning the fusion weights. ECF directly relates the performance of the system to the fusion parameters and determine the weights one at a time iteratively by solving a 2-class, one-dimensional problem to minimize the errors of the classifier fusion. We expected that with the non-linearity and error correcting capabilities of the ECF learning algorithm, further discriminative information in the non-linear subspaces at the score level is employed to correct the errors of the classifier fusion during the learning process.

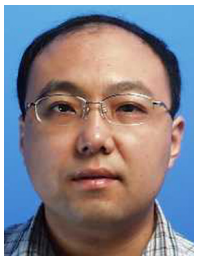
To validate the effectiveness of the ECF method, we used multiple output scores generated from 7 language classifiers in the IIR's submission to the 2009 NIST LRE evaluation. The experiments were conducted on the 2009 NIST LRE 30-, 10-, and 3-sec evaluation sets. We also compared the ECF method to several other fusion methods. The experimental results demonstrated the effectiveness of the ECF system to improve the performance of the best individual sub-system. The results also showed that the ECF system outperforms the comparable methods such as the FoCal system. Finally, we showed that by incorporating sub-systems with complementary information in the fusion system, the number of parameters of the fusion system to achieve the optimal performance can be reduced.

References

[1] Y.K. Muthusamy, E. Barnard, and R.A. Cole, "Reviewing automatic language identification," IEEE Signal Process. Mag., vol.11, no.4, pp.33–41, 1994.
[2] 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, 1996.
[3] H. Li, B. Ma, and C.H. Lee, "A vector space modeling approach to spoken language identification," IEEE Trans. Audio, Speech Language Processing, vol.15, no.1, pp.271–284, 2007.
[4] P.A. Torres-Carrasquillo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, and J.R. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," Proc. Internat. Conf. Spoken Lang. Process., pp.89–92, 2002.
[5] W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, and P.A.

Torres-Carrasquillo, "Support vector machines for speaker and language recognition," Comput. Speech Lang., vol.20, pp.210–229, 2006.
[6] Y. Obuchi and N. Sato, "Language identification using phonetic and prosodic HMMs with feature normalization," Proc. Internat. Conf. on Acous. Speech Signal Process., pp.569–572, 2005.
[7] L. Mary and B. Yegnanarayana, "Extraction and representation of prosodic features for language and speaker recognition," Speech Commun., vol.50, pp.782–796, 2008.
[8] D. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," Digit. Signal Process., vol.10, pp.19–41, 2000.
[9] W. Campbell, R. Gleason, D. Reynolds, W. Shen, E. Singer, and P. Torres-Carrasquillo, "Advanced language recognition using cepstras and phonotactics: MITLL system performance on the NIST 2005 language recognition evaluation," IEEE Odyssey, 2006.
[10] S. Nakagawa, Y. Ueda, and T. Seino, "Speaker-independent, text-independent language identification by HMM," Proc. Internat. Conf. Spoken Lang. Process., vol.2, pp.1011–1014, 1992.
[11] F.S. Richardson and W.M. Campbell, "Language recognition with discriminative keyword selection," Proc. Internat. Conf. on Acous. Speech Signal Process., pp.4145–4148, 2008.
[12] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," Proc. IEEE Odyssey, pp.219–226, 2004.
[13] W.M. Campbell, D.E. Sturim, P.A. Torres-Carrasquillo, and D.A. Reynolds, "A comparison of SubSpace feature-domain methods for language recognition," Proc. Interspeech, pp.309–312, 2008.
[14] W.M. Campbell, D.E. Sturim, and D.A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," IEEE Signal Process. Lett., vol.13, no.5, pp.308–311, 2006.
[15] http://www.itl.nist.gov/iad/mig/tests/lre/
[16] T. Rong, B. Ma, Z. Donglai, H. Li, and E.C. Chng, "Integrating acoustic, prosodic and phonotactic features for spoken language identification," Proc. Internat. Conf. on Acous. Speech Signal Process., pp.205–208, 2006.
[17] 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," Proc. EuroSpeech, pp.1345–1348, 2003.
[18] P.A. Torres-Carrasquillo, E. Singer, W. Campbell, T. Gleason, A. McCree, D.A. Reynolds, F. Richardson, W. Shen, and D. Sturim, "The MITLL NIST LRE 2007 language recognition system," Proc. Interspeech, pp.22–26, 2008.
[19] J. Bezdek, Fuzzy models and algorithms for pattern recognition and image processing, Kluwer Academic, 1999.
[20] N. Brummer and D. Leeuwen, "On calibration of language recognition scores," Proc. IEEE Odyssey-Speaker Lang. Recognition Workshop, pp.1–8, 2006.
[21] L. Ferrer, E. Shriberg, S.S. Kajarekar, A. Stolcke, K. Sonmez, A. Venkatarman, and H. Bratt, "The contribution of cepstral and stylistic features to SRI's NIST speaker recognition evaluation system," Proc. Internat. Conf. on Acous. Speech Signal Process., pp.101–104, 2006.
[22] J.P. Campbell, D.A. Reynolds, and R.B. Dunn, "Fusing high- and low-level features for speaker recognition," Proc. Eurospeech, pp.2665–2668, 2003.
[23] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," Proc. Eurospeech, pp.1895–1898, 1997.
[24] http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf
[25] N. Brummer and J. Preez, "Application-independent evaluation of speaker detection," Comput. Speech Lang., vol.20, pp.230–275, 2006.
[26] H. Li, B. Ma, K.A. Lee, H. Sun, D. Zhu, K.C. Sim, C. You, R. Tong, I. Krkkinen, C.L. Huang, V. Pervouchine, W. Guo, Y. Li, L. Dai, M. Nosratighods, T. Tharmarajah, J. Epps, E. Ambikairajah, E.C. Chng, T. Schultz, and Q. Jin, IIR System Description for the 2009 NIST Language Recognition Evaluation, the 2009 NIST LRE workshop, Baltimore, USA, 2009.
[27] C.H. You, K.A. Lee, and H. Li, "An SVM kernel with GMM supervector based on the Bhattacharyya distance for speaker recognition," IEEE Signal Process. Lett., vol.16, no.1, pp.49–52, 2009.
[28] C.H. You, K.A. Lee, and H. Li, "A GMM supervector kernel with the bhattacharyya distance for SVM based speaker recognition," Proc. Internat. Conf. on Acous. Speech Signal Process., pp.4221–4224, 2009.
[29] W.M. Campbell, "A covariance kernel for SVM language recognition," Proc. Internat. Conf. on Acous. Speech Signal Process., pp.4141–4144, 2008.
[30] D. Zhu, B. Ma, and H. Li, "Large margin estimation of Gaussian mixture model parameters with extended Baum-Welch for spoken language recognition," Proc. INTERSPEECH, pp.2179–2182, 2009.
[31] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus Eigenchannels in speaker recognition," IEEE Trans. Audio, Speech, Language Process., vol.15, no.4, pp.1435–1447, 2007.
[32] K.A. Lee, C.H. You, K.C. Sim, and H. Li, "Expected likelihood and fast computation of posterior weights for spoken language recognition," Proc. INTERSPPECH, 2009.
[33] B. Ma, H. Li, and R. Tong, "Spoken language recognition using ensemble classifiers," IEEE Trans. Audio, Speech Language Process., vol.15, no.7, pp.2053–2062, 2007.
[34] R. Tong, B. Ma, H. Li, and E.C. Chng, "Target-aware language models for spoken language recognition," Proc. INTERSPEECH, pp.200–203, 2009.
[35] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," Technical Report, HPL-2004, 2004.
[36] K. Woods, W.P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.4, pp.405–410, 1997.
[37] A. Altmann, M. Rosen-Zvi, M. Prosperi, E. Aharoni, H. Neuvirth, E. Schulter, J. Buch, D. Struck, Y. Peres, F. Incardona, A. Sonnerborg, R. Kaiser, M. Zazzi, and T. Lengauer, "Comparison of classifier fusion methods for predicting response to anti HIV-1 therapy," PLoS ONE, 3, e3470, 2008.
[38] V. Vapnik, The Nature of Statistical Learning Theory, Springer, N.Y., 1995.

Omid Dehzangi received his B.Eng. and M.Eng. degrees in computer engineering from Shiraz University, Iran, 2004 and 2007. He is currently pursuing his Ph.D. degree in computer engineering at Nanyang Technological University, Singapore. His research interests include pattern recognition and discriminative analysis, speaker and spoken language recognition, brain-computer interface.



Bin Ma (*S M′06*) received the B.Sc. degree from Shandong University, Jinan, China, in 1990, the M.Sc. degree i from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, in 1993, and the Ph.D. degree in computer engineering from The University of Hong Kong in 2000. He was a Research Assistant from 1993 to 1996 at National Laboratory of Pattern Recognition, IACAS. In 2000, he joined Lernout and Hauspie Asia Pacific as a Researcher focusing on the speech recognition of multiple Asian languages. From 2001 to 2004, he worked for InfoTalk Corporation, Ltd., and became the Senior Technical Manager engaging in mix-lingual telephony speech recognition system for the Asia-Pacific market, and in embedded speech recognition system on PDAs and hand-phones. Since 2004, he has been a Research Scientist and Group Leader of the Speech and Dialogue Processing Group, Institute for Infocomm Research, Singapore. His current research interests include robust speech recognition, speaker and language recognition, spoken document retrieval, natural language processing, and machine learning.

Eng Siong Chng (*S M′05*) received the B.Eng. degree in electrical and electronics engineering and the Ph.D. degree from the University of Edinburgh, Edinburgh, U.K., in 1991 and 1996, respectively. He is currently an Assistant Professor in the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. Prior to joining NTU in 2003, he was with the Institute of Physics and Chemical Research, Riken, as a Postdoctoral Researcher working in the area of signal processing and classification (1996), the Institute for Infocomm Research (IIR) as a member of research staff to transfer the Apple-ISS speech and handwriting technologies to ISS (1996–1999), Lernout and Hauspie as a Senior Researcher in speech recognition (1999–2000), and Knowles Electronics as a Manager for the Intellisonic microphone array research (2001–2002). His research interests are in pattern recognition, signal, speech, and video processing. He has published over 50 papers in international journals and conferences. He is currently leading the speech and language technology program (http://www3.ntu.edu.sg/home/aseschng/SpeechTechWeb/default.htm) in the Emerging Research Lab at the School of Computer Engineering, NTU.

Haizhou Li (*S M′01*) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and electronic engineering from the South China University of Technology (SCUT), Guangzhou, in 1984, 1987, and 1990, respectively. He was a Research Assistant from 1988 to 1990 at the University of Hong Kong. In 1990, he joined SCUT as an Associate Professor where he became a Full Professor in 1994. From 1994 to 1995, he was a Visiting Professor at the Nancy Research Centre of Computer Science (CRIN), Nancy, France. In 1995, he became the Manager of the ASR Group at the Apple-ISS Research Centre in Singapore where he led the research of Apples Chinese Dictation Kit for Macintosh. In 1999, he was appointed as the Research Director of Lernout and Hauspie Asia Pacific. From 2001 to 2003, he was the Vice President of InfoTalk Corp., Ltd., in Singapore. Since 2003, he has been with the Institute for Infocomm Research (IIR) in Singapore, where he is now the Principal Scientist and Department Head of Human Language Technology, and the Program Manager of Social Robotics. He is also a Visiting Professor (Honorary) of the School of Electrical Engineering and Telecommunications at University of New South Wales, Australia; Nokia Visiting Professor 2009, Nokia Foundation, Finland. His current research interests include automatic speech recognition, speaker recognition, spoken language recognition, and natural language processing. Dr Li was a recipient of the National Infocomm Award 2001 and the TEC Innovators Award 2004 in Singapore. He now serves as an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and the Springer International Journal of Social Robotics. He is also a Vice President of the COLIPS, a Board Member of International Speech Communication Association (ISCA), and a Board Member of the Asian Federation of Natural Language Processing (AFNLP). He was the Local Chair of SIGIR 2008 and ACL-IJCNLP 2009.