Instance Based Sparse Classifier Fusion for Speaker Verification

Mohammad Hasheminejad
Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran
mhashemi@birjand.ac.ir
Hassan Farsi*
Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran
hfarsi@birjand.ac.ir

Received: 28/Feb/2016 Revised: 26/Apr/2016 Accepted: 07/May/2016

Abstract

This paper focuses on the problem of ensemble classification for text-independent speaker verification. Ensemble classification is an efficient method to improve the performance of the classification system. This method gains the advantage of a set of expert classifiers. A speaker verification system gets an input utterance and an identity claim, then verifies the claim in terms of a matching score. This score determines the resemblance of the input utterance and pre-enrolled target speakers. Since there is a variety of information in a speech signal, state-of-the-art speaker verification systems use a set of complementary classifiers to provide a reliable decision about the verification. Such a system receives some scores as input and takes a binary decision: accept or reject the claimed identity. Most of the recent studies on the classifier fusion for speaker verification used a weighted linear combination of the base classifiers. The corresponding weights are estimated using logistic regression. Additional researches have been performed on ensemble classification by adding different regularization terms to the logistic regression formulae. However, there are missing points in this type of ensemble classification, which are the correlation of the base classifiers and the superiority of some base classifiers for each test instance. We address both problems, by an instance based classifier ensemble selection and weight determination method. Our extensive studies on NIST 2004 speaker recognition evaluation (SRE) corpus in terms of EER, minDCF and minCLLR show the effectiveness of the proposed method.

Keywords: Speaker Recognition; Speaker Verification; Ensemble Classification; Classifier Fusion; IBSparse**.**

1. Introduction

Scientific studies have shown that, there are varieties of information in a speech signal which can help speaker recognition. Speaker recognition is a process of decision making about a speaker's identity using the person's speech signal. The field of speaker recognition contains two main branches; speaker verification and speaker identification. In speaker verification, an identity claim is first constructed and then the claim is accepted, or rejected, based on the information extracted from the corresponding speech signal. On the other hand, a speaker identification system, at first, registers a set of target speakers and then determines the identity of the owner of an incoming speech signal. Since a speaker verification system can lead to speaker identification and there are more sophisticated criteria to evaluate a speaker verification system, the majority of speaker recognition research is devoted to speaker verification tasks. To gain advantage of different information of speech in the verification process, an ensemble of base classifiers can be used. Classifier fusion is an important subject in speaker verification which can be performed on the feature, score or decision level [1]. On the feature level fusion, different feature vectors are concatenated to construct a new feature vector. In speaker verification, fusion of scores includes obtaining matching scores for each base classifier and obtaining a final score from these base scores using a proper role. On the decision level fusion, the final decision is a logical fusion of decision output of different classifiers or modalities. This logical fusion can be *"AND", "OR"* or a combination of both. In this paper, we focus on score level fusion where the final score is a weighted summation of base scores. Contrary to most of the classifier fusion for speaker verification works, which use the weighted sum of scores for all test instances [2], or a simple arithmetic mean of scores as the final score [3], we use an instance-specific ensemble of classifiers whose weights are adopted separately for each test instance. Despite that using permanent weights for score fusion in a speaker verification task, may be effective in some situations, obtaining optimum weights which are effective for all test instances is troublesome. In these methods a set of unique weights are learned on training or held back data. Thereafter, these weights are used in the verification process of all trials. This may not be generalizable to all test samples, since some base classifiers may be effective for some of the test samples and not for others. In this paper, we consider this issue and exploit the instance-specific behavior speaker classifiers. To do this we were inspired by [2], [4] and [5], and act in the following procedures:

* Corresponding Author

- We determine the weight of each classifier according to the test instance. Then, we calculate the final score as a weighted sum of scores obtained from all base classifiers.
- We also consider sparse classifier fusion using the behavior. Therefore, in this case, the final score is a weighted sum of scores from a few base classifiers.
- We introduce a new formula to determine the final fusion score.

Logistic regression with elastic-net regularization is also considered as a baseline to show the effectiveness and generalization power of the proposed method.

2. Base Classifiers

A typical speaker verification system consists of train and test phases. In the train phase we introduce target speakers to the system. In the test phase incoming unlabeled utterances are claimed as belonging to one of the enrolled targets and the system verifies validity of this claim. Figure 1 shows workflow of such a system. Speaker feature extraction methods transform the original speech signal to a compact representation. These methods aim at holding speaker specific information it the resulting representation.

To create powerful base classifiers, we used four widely used speech features in speaker recognition. These features are mel-frequency cepstral coefficients (MFCC), perceptual linear prediction (PLP), stabilized weighted linear prediction (SWLP) [6], and linear predictive cepstral coefficient (PLCC) [7]. Conventional linear prediction (LP) determines a pth order autoregressive model for a speech frame, by minimizing the sum of squares of prediction errors. Weighted linear prediction (WLP) is obtained by introducing temporal weighting of the squared prediction error. The SWLP which is used in our research, is a variant of WLP that guarantees the stability of WLP filter. A concise description of MFCC and PLP feature extraction is provided in [8].

In the matching step, the system tries to specify the similarity of enrolled target features (templates) and incoming speech features, in terms of a verification score. In the late steps of the verification, the score is compared to a predefined threshold value. This threshold value is computed from training data. If the score of the trial is more than the threshold, the claim is accepted, otherwise it is rejected.

We used three different powerful modeling methods of speakers. These methods are, GMM-SVM-KL [9], GMM-SVM-BHAT [10] and ivector-PLAD [11]. SVM based methods have been successful in text independent speaker verification. In these methods, the SVM is combined with the GMM supervector concept. They derive a kernel (we used Kullback-Leibler (KL) divergence and Bhattacharya (BHAT) distance), then apply the Nuisance Attribute Projection (NAP) [12] to the kernel. Total variability or ivector systems provide an elegant way of dimensionality reduction of speech features. This technique converts a sequence of feature frames to a fixed length low dimensional vector. This vector represents the whole utterance (i.e. the whole speaker) and can be an input to a standard pattern recognition algorithm. We then use Probabilistic Linear Discriminant Analysis (PLDA) for scoring.

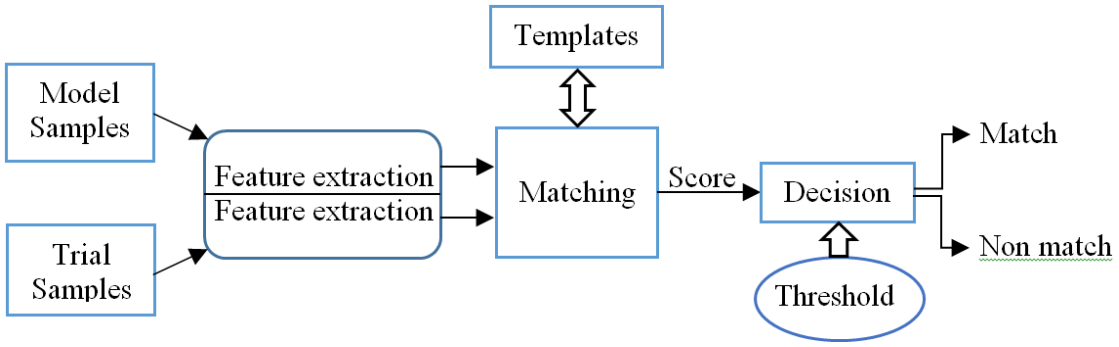

Fig. 1. Block diagram of a typical speaker verification system

3. Score Fusion in Speaker Verification

There are three main levels on which biometric classifier fusion can occur: feature level fusion, score level fusion and decision level fusion. Feature level fusion (early fusion methods) occurs before the invocation of the matching block (Figure 1). In this process a new feature vector is created. This new feature vector is a combination of previously extracted features. The matching step is performed on the new feature vector. In score level fusion (late fusion methods), which is our main subject, some basis expert classifiers are firstly employed to obtain the matching score between each test sample, and the previously stored templates. It is shown that score level fusion methods provide better results than feature level fusion [5]. In decision level fusion approaches, accept or reject decision of individual classifiers serve as input to the fusing function [13].

In the case of score fusion, the final score is obtained as, $s_f = w_0 + \sum_{l=1}^{L} w_l s_l$, in which L is number of base classifiers, w_0 is added to calibrate the final score and w_l and s_l are weight and score of l^{th} classifier, respectively. Regardless of training individual classifiers, there is a need to train the fusion process. An intuitive way of obtaining a final score from ensemble classifier scores is to estimate the fixed weight for each classifier. To do this one needs a set of labeled scores, whose labels are either 0 ($y_i = 0$) if the utterance belongs to the claimed target or 1 ($y_i = 1$) if the utterance does not belong to the claimed target. If the number of training scores is N_{dev}, a development set $D = \{(s_i, y_i, I = 1, \dots, N_{dev})\}$ is used to train this model. Such a system does not need to know anything about how individual classifiers are trained or the speech features. After selecting proper training scores an optimization method is employed to minimize an error criterion or maximize an efficiency measure. The optimization can be directly performed using a neural network [14], heuristic algorithms [15] or the widely used logistic regression [2].

3.1 Logistic Regression Based Fusion

State-of-the-art speaker verification systems use multiple classifiers to make a reliable decision. Linear regression is a discriminative model [16] which is commonly used to fuse scores in speaker verification. In this section we explain why this method is widely used and accepted in speaker verification and how it is improved in recent years.

In test phase of an ensemble speaker verification system, an identity is firstly claimed for an incoming utterance signal. Each classifier in the ensemble, measures validity of the claim, in terms of similarity score. At this stage the system needs a score fusion method to accept or reject the claim. The score fusion method should realize a mapping from $\mathbb{R}^n$ space to a binary space $\{0, 1\}$, where 0 means the identity claim is accepted and 1 means it is rejected. We can cast the problem as two target classification problems with an n dimensional input feature vector. Elements of these vectors should be of the same type, e.g. probability. One may calculate best weights, which minimize classification error for the training set, using a brute force approach. But, there is a question of generalization. There is considerable variation between training and runtime scores. This is why it is recommended to use estimates of real probabilities as scores [17]. Bayesian framework [16], which minimizes classification error probability, can be used to reach those probabilities. We will provide a general overview of the Bayesian decision rule next here.

Suppose there are two classes, T and I, representing target and non-target (Imposter) classes, respectively. For a given random score vector of X, which may belong to either of class the cost of classifying a class i score vector into a class j event, can be a zero-one loss function (Equation (1)):

$$\mathrm{C_{ij}} = \begin{cases} 0 & \text{if i} = \text{j} \\ 1 & \text{if i} \neq \text{j} \end{cases} \quad (1)$$

This assigns zero loss to a correct classification and a unit loss to a miss classification. Under this assumption, Bayes rule defines the posterior probability of class i as Equation (2):

$$\mathrm{P(c_i|X)} = \frac{\mathrm{P(X|c_i).P(c_i)}}{\mathrm{P(X)}} \quad (1)$$

Where $P(X)$ is prior probability of X and $P(c_i)$ is prior probability of c_i. We need to estimate probability distribution correctly, to get reliable scores. In [17] it is explained how we can reduce Equation (2) to $P(X|c_i)$ using assumptions about prior probabilities. If we suppose different base classifiers are independent of one another, $P(X|c_i)$ results in equation (3):

$$\mathrm{P(X|c_i)} = \mathrm{P(s_1, \dots, s_K|c_i)} = \prod_{\mathrm{k=1}}^{\mathrm{K}} \mathrm{P(s_k|c_i)} \quad (2)$$

Where K is the number of base classifiers, c_i is a label for class i and s_k is the k^{th} element of score vector. Since the scores for T class (target) are correlated, the recent assumption is not intuitive. This is due to the fact that if a trial belongs to a target class, scores of all good classifiers are close to unity. It is more reasonable to believe that s_k for the imposter and $(1 - s_k)$ for the target are not correlated. The posterior probability of target class can be derived as equation (4) [17]:

$$\mathrm{P(T|s_1, \dots, s_k)} = \frac{1}{1 + \mathrm{e}^{-\{(\sum_{\mathrm{k=1}}^{\mathrm{K}} \mathrm{x_k}) + \mathrm{x_0}\}}} \quad (3)$$

Where $\mathrm{x_0} = \ln \frac{\mathrm{P(T)}}{\mathrm{P(I)}}$, and $\mathrm{x_k} = \ln \frac{\mathrm{P(s_k|T)}}{\mathrm{P(s_k|I)}}$.

If we suppose that the probabilities are members of the exponential family (equations (5) and (6)):

$$\mathrm{P(s_k|T)} = \mathrm{f(s_k)}.\mathrm{e}^{(\mathrm{C_k}.\mathrm{s_k} + \mathrm{C_{k0}})} \quad (4)$$

$$\mathrm{P(s_k|I)} = \mathrm{f(s_k)}.\mathrm{e}^{(\mathrm{C_k}.\mathrm{s_k} + \mathrm{C_{k0}})} \quad (5)$$

Then equation (4) is reduced to logistic regression (LR) model or logistic distribution function (equation (7)):

$$\mathrm{P(T|s_1, \dots, s_k)} = \frac{1}{1 + \mathrm{e}^{-\mathrm{g(s)}}} = \pi \quad (6)$$

Where:

$$\mathrm{g(s)} = \beta_0 + \beta_1.\mathrm{s_1} + \cdots + \beta_{\mathrm{K}}.\mathrm{s_K} \quad (7)$$

$$\beta_0 = \sum_{\mathrm{k=1}}^{\mathrm{K}} (\mathrm{C_{k0}} - \mathrm{I_{k0}}) + \ln \frac{\mathrm{P(T)}}{\mathrm{P(I)}} \quad (8)$$

$$\beta_{\mathrm{K}} = \mathrm{C_k} + \mathrm{I_k} \quad (9)$$

A particular case of the exponential family is a Gaussian distribution. If we suppose distribution of the classes are Gaussian, equations (9) and (10) become equal to equations (11) and (12).

$$\beta_0 = \sum_{k=1}^{K} \frac{(\mu_k^I)^2 - (\mu_k^T)^2}{2\sigma_k^2} + \ln \frac{P(T)}{P(I)} \quad (10)$$

$$\beta_K = \frac{\mu_k^T - \mu_k^I}{\sigma_k^2} \quad (11)$$

Where μ_k^T and μ_k^I are the mean of the target and imposter distributions, respectively, and σ_k^2 is the common variance. An interesting result of this method is that, the weight of the k[th] classifier, β_K, is proportional to the difference of the means of, the target and imposter distributions and if the classifier has scattered target scores it is not reliable and has a lower weight.

To this point we should find optimal weights (β_K in equation (8)), so that $P(T|s_1, \dots, s_k)$, indicated by equation (7), is maximized. To solve the problem, researchers took many issues in to consideration and introduced cost functions. One of the most recent defined cost functions is $C_{wlr}(w, D)$[2] which is given by:

$$\mathrm{C_{wlr}(w)} = \frac{\mathrm{P_{eff}}}{\mathrm{N_t}} \sum_{\mathrm{i=1}}^{\mathrm{N_t}} \log\left(1 + \mathrm{e}^{-\mathrm{w^T s_i} - \mathrm{logit\, P_{eff}}}\right) + \frac{1 - \mathrm{P_{eff}}}{\mathrm{N_f}} \sum_{\mathrm{j=1}}^{\mathrm{N_f}} \log\left(1 + \mathrm{e}^{\mathrm{w^T s_j} + \mathrm{logit\, P_{eff}}}\right) \quad (12)$$

Where $P_{eff} = logit^{-1}\left(logit(P_{tar}) + log\left(C_{miss}/C_{fa}\right)\right)$, depends on the prior probability of a target speaker (P_{tar}), the cost of miss classification (C_{miss}) and the cost of false acceptation (C_{fa}). The aim of defining such a cost function is to find the optimal weights which minimize the cost function. Equation (14) formulates this optimization problem:

$$\mathrm{w^*} = \underset{\mathrm{w}}{\operatorname{argmin}}\, \mathrm{C_{wlr}(w)} \quad (13)$$

This formulation changed to (15) when V. Hautamäki, et.al. showed in [2] that, a regularized version of equation (14), which takes a sparse number of classifiers in the ensemble, acts better. This optimization problem is regularized using a combination of ridge and LASSO regressions which is called elastic-net.

$$\mathrm{w^*} = \underset{\mathrm{w}}{\operatorname{argmin}}\{\mathrm{C_{wlr}(w)} + \lambda(\alpha\|\mathrm{w}\|_1 + (1-\alpha)\|\mathrm{w}\|_2^2)\} \quad (14)$$

Where coefficient, λ, which is a Lagrange multiplier, determines the amount of shrinkage of the weights. The constraint $\|w\|_1$ is known as LASSO and $\|w\|_2^2$ corresponds to ridge regression.In elastic-net, the LASSO part causes most of the weights to be near zero. This means that, it is a sparsity promoting constraint. The other part of the elastic-net is ridge constraint, which causes the weights not to push as aggressively as LASSO only constraint. Coefficient α determines the amount of participation of the LASSO and ridge in the equation. This problem can be solved using the ProjectL1 algorithm [18][1].

Although this method tries to increase the generalization capability of the classifier fusion, it identifies a unique weight for every classifier. These weights are obtained from training or held-out data and are used on all instances. This method also chooses a sparse number of classifiers during the training process and omits most of them from the test process. We show that a classifier, which is omitted from the ensemble set, has better performance for specific test instances. Recent studies showed that, taking the instance based behavior of classifiers improves generalization of the ensemble classifier [4],[5]. In the following section we introduce the proposed method to take the instance base behavior of speaker verification experts.

3.2 Instance Based Ensemble and Weight Selection

Weights which are selected based on a test sample, should score the prediction capability of each classifier on that sample. If the test sample in a trial is *positive* (real target), and the score of a classifier is high, then its weight should be high, and the weight should be low when this score is low. If the sample is *negative* (is not target), and the score of a classifier is high its weight should be low and if its score is low, it means that the classifier has made a reasonable decision, and the weight should be high. We call the proposed method instance based sparse classifier fusion (IBSparse).

Discovering individual weights for each trial is a challenging task. Since there is no information about the real label of the trial, we do not know if the classifier decision is correct or not, and as a result, it is not clear how to derive a specific weight for the classifier.

3.2.1 Clarity Index

Clarity index is an objective that can be used to obtain sample specific weights [4]. This objective is based on test scores and previously obtained training scores and has nothing to do with low level features. Each classifier has n_0 positive and n_1 negative training scores, which are obtained in the training phase. The positive scores are those scores whose related utterance originated from the target and negative scores are scores that belong to the impostor utterances. The Clarity index depends on two factors. The first factor is Relevance Loss (RL) which determines the position of a test score vector, S^{ts}, against negative training scores (S_i^{ntr}). Equation (16) defines RL:

$$\mathrm{RL(S^{ts}, W)} = \frac{1}{\mathrm{n_n}} \sum_{\mathrm{i=1}}^{\mathrm{n_n}} \mathrm{U(W^T S_i^{ntr} - W^T S^{ts})} \quad (15)$$

Where W is the weight vector, S^{ts} is the test score vector, n_n is the number of negative training scores and U is the unit step function. RL is a fraction of the non-target

[1] Available online at: "http://www.cs.ubc.ca/~schmidtm/Software/code.html"

training scores divided by the total number of non-target scores. Therefore, this value is in the range of [0,1]. For a target trial, the ideal state is that the test score will be higher than all negative training scores. As a result, this value is desired to be close to 0. For a non-target trial, the ideal state is that the test score would be less than all negative training scores, consequently, the value is desired to be close to one.

The second factor is irrelevance loss (IL) which determines the position of a test score vector against the positive training scores (S_i^{ptr}). Equation (17) defines IL:

$$\mathrm{IL}(\mathrm{S^{ts}},\mathrm{W}) = \frac{1}{\mathrm{n_p}}\sum_{\mathrm{j=1}}^{\mathrm{n_p}} \mathrm{U}\left(\mathrm{W^T S^{ts}} - \mathrm{W^T S_j^{ptr}}\right) \qquad (16)$$

Where n_p is the number of positive training scores. IL is desired to be close to 1 for a target trial and 0 for a non-target trial.

The raw clarity index is then defined as the difference between RL and IL (Equation (18)):

$$\mathrm{RCL}(\mathrm{S^{ts}},\mathrm{W}) = \mathrm{RL}(\mathrm{S^{ts}},\mathrm{W}) - \mathrm{IL}(\mathrm{S^{ts}},\mathrm{W}) \qquad (17)$$

An absolute value of RCL it called the clarity index (Equation (19)):

$$\mathrm{CL}(\mathrm{S^{ts}},\mathrm{W}) = |\mathrm{RL}(\mathrm{S^{ts}},\mathrm{W}) - \mathrm{IL}(\mathrm{S^{ts}},\mathrm{W})| \qquad (18)$$

The range of the clarity index is [0,1]. As it is mentioned, for a target trial the ideal value of RL is 0 and IL is 1, thus the ideal value of CL is 1. For a non-target trial, the ideal value of CL is also 1. Thus a higher value for the clarity index means that the decision is more dependable. Therefore we use it to select a sparse number of classifiers and use it in the weight learning process.

3.2.2 Weight Learning and Ensemble Selection

By using the clarity index in the classifier selection we solve three problems. The first problem is as a result of the fact that classifiers have different performance with respect to different test samples. This in fact, affects both classifier selection and weight determination, which is not exploited in previous works. The second problem is in choosing an efficient number of classifiers and the third is the correlation between the classifiers. There may be a different number of efficient classifiers for different test samples and they may or may not correlate in different situations. In sparse classifier fusion for speaker verification, these problems are not efficiently addressed either. By using the clarity index and a proper threshold value, we can choose an adaptive number of efficient classifiers. The proper threshold value is chose from the training scores so that the final EER for the training scores is minimized. In the case where the clarity index of all classifiers falls below the threshold, we use a predefined minimum number of classifiers.

In the ensemble selection process we do not have the weight vector to calculate the clarity index for each classifier. Thus, we change RL and IL formulation and replace $W^T S_i^{ntr}$ and $W^T S_j^{ptr}$ with s_i^{ntr} and s_j^{ptr} respectively. $s_i^{ntr}, i = 1, \dots, n_n$ are negative scalar scores and $s_i^{ptr}, i = 1, \dots, n_p$ are positive scalar scores related to the classifier.

We use two strategies for sample based classifier ensemble selection. In the first scenario, we choose a fixed threshold on the clarity index. Classifiers with a clarity index higher than the threshold are used in the ensemble. With this strategy, different classifiers are used for different test samples. In the case where all the indices are lower than the threshold, all 12 classifiers participated in the ensemble. In the second strategy, the threshold is not fixed and varies according to the values of the clarity index, related to each test sample. In this scenario, the test score of each classifier is first calibrated to log the likelihood ratio [19] then we use the threshold to select confident classifiers.

To take the sample specific behavior of classifiers and gain the generalization ability of equation (15) we propose to use equation (20):

$$\underset{\mathrm{W}}{\mathrm{argmin}}\, \mathrm{C_{wlr}}(\mathrm{W}) + \lambda\mathcal{F}(\mathrm{S^{ts}},\mathrm{S^{tr}}) \qquad (19)$$

Where $\mathcal{F}$ is a function of the test sample, current weights, and positive and negative training samples. If we directly substitute CL into the equation (20), the optimization becomes generally intractable, because due to the definition of RL and IL it is a discrete measure and cannot be differentiated. Thus we approximate the discrete relevant and irrelevant losses by differentiable sigmoid functions (Equations (21) and (22)):

$$\mathrm{RL}(\mathrm{S^{ts}},\mathrm{W}) = \frac{1}{\mathrm{n_n}}\sum_{\mathrm{i=1}}^{\mathrm{n_n}} \frac{1}{1+\mathrm{e}^{-\alpha \mathrm{W^T}(\mathrm{S_i^{ntr}} - \mathrm{S^{ts}})}} \qquad (20)$$

$$\mathrm{IL}(\mathrm{S^{ts}},\mathrm{W}) = \frac{1}{\mathrm{n_p}}\sum_{\mathrm{j=1}}^{\mathrm{n_p}} \frac{1}{1+\mathrm{e}^{-\beta \mathrm{W^T}(\mathrm{S^{ts}} - \mathrm{S_j^{ptr}})}} \qquad (21)$$

By choosing the correct value for α and β these two equations can be close to the original values of RL and IL. Setting a high value for α and β results in a closer approximation to the true values of RL and IL, but, results in several local optima for CL. On the other hand, a low value of these two parameters may result in a poor approximation of CL. Consequently these parameters have considerable effect on the performance of the classification. Even with this modification we still do not substitute CL into equation (20) because it is an absolute value of difference between RL and IL, and is not differentiable at zero. To get rid of this we use the ridge regression [16] of the raw clarity index (RCL).
Finally we define the optimization problem as:

$$\underset{\mathrm{W}}{\mathrm{argmin}}\, \mathrm{C_{wlr}}(\mathrm{W}) - \lambda\|\mathrm{RCL}\|_2^2 \qquad (22)$$

Although the optimization process is performed in the test phase, it is fairly fast and in less than half a second converges to the optimal points. This problem can be solved using standard packages [16]. For the case of faster optimization, we propose to optimize weights for

all possible combinations of classifiers, and use proper weights after the ensemble selection using CL. Using this method, at first, we determine all combinations of classifiers. If we have n classifiers, the number of these combinations is $\sum_{i=1}^{n} \frac{n!}{(n-i)!i!}$. There are 4095 combinations when we have 12 base classifiers. Then, equation (24) is solved for each combination separately to obtain a table of weight vectors:

$$\underset{w}{\operatorname{argmin}}\, C_{wlr}(W) - \lambda \|w\|_2^2 \qquad (23)$$

In this equation, ridge regularization keeps weights small.

In the test process the score of each classifier is first calibrated. Then the clarity index is computed for each calibrated score, and confident classifiers are selected using the clarity index. Finally, confident scores are fused using related weights. Figure 2 depicts the block diagram of proposed ensemble classification system.

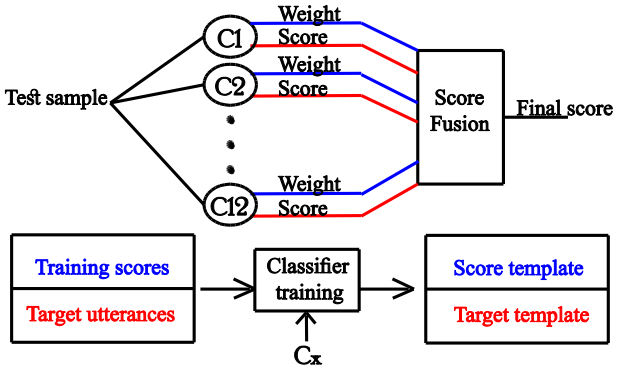

Fig. 2. Block diagram of the proposed ensemble classification system. Each of the classifiers C1-C12 contain score templates.

4. Experimental Results

4.1 Databases

We used NIST 2004 Speaker Recognition Evaluation (SRE), and switchboard II in our experiments. Since we use many classifiers and each classifier or feature has an ability to detect special characteristics of a speech signal, we preferred not to restrict training or test data to originating from a male or female, or specific language. NIST 2004 contains 6244 training files. The Universal background model is trained on these data. This dataset also contains 660 male and female speakers, and 4623 test utterances. These utterances are from five different languages: Arabic, English, Mandarin, Russian and Spanish. In the case in which an utterance has more than one minute duration we split the utterance to have more test data. Switchboard II (2348 conversation sides) is also used to train the PLDA dimensionality reduction process, λ, and nuisance attribute projection (NAP).

4.2 Experimental Setup

It is believed that diversity of base classifiers improves the performance of ensemble classification [20]. In addition, features used and the methods of classification should be efficient enough for the classification. Therefore, our experiments are conducted on three well-known different classifiers and four different feature vectors. We used MFCC, PLP, SWLP, and PLCC as different speech features.

Table 1. Twelve different base classifiers implemented on NIST04 dataset, using four different features and three methods plus fusion systems. Proposed spample based (Instance basced) methos spesified as IBSparse1&2

	Classifier	Feature	Devset			Evaset		
			EER(%)	MinDCF ×100	MinCLLR ×100	EER(%)	MinDCF ×100	MinCLLR ×100
1	Ivector-PLDA	MFCC	6.68	4.68	21.78	8.01	5.59	24.59
2	Ivector-PLDA	LPCC	7.25	5.74	23.39	5.79	4.85	19.44
3	Ivector-PLDA	PLP	6.59	5.06	21.59	7.76	5.27	23.92
4	Ivector-PLDA	SWLP	9.12	8.59	30.73	10.65	8.37	34.48
5	GMM-SVM-BHAT	MFCC	7.23	7.23	25.30	8.12	7.76	24.76
6	GMM-SVM-BHAT	LPCC	8.35	6.01	25.27	7.59	6.16	23.35
7	GMM-SVM-BHAT	PLP	8.15	6.67	25.08	7.71	6.30	22.01
8	GMM-SVM-BHAT	SWLP	10.54	8.19	30.53	11.05	8.48	29.54
9	GMM-SVM-KL	MFCC	7.44	5.53	23.78	9.12	6.71	26.94
10	GMM-SVM-KL	LPCC	6.66	4.69	23.98	8.41	5.67	25.40
11	GMM-SVM-KL	PLP	7.45	6.42	25.02	8.37	6.91	25.27
12	GMM-SVM-KL	SWLP	7.88	5.56	27.45	9.47	6.57	29.07
13	Sparse_fusion	-	**3.05**	**2.75**	**10.36**	3.37	3.02	11.63
14	IBSparse1	-	-	-	-	**2.56**	2.26	**8.01**
15	IBSparse2	-	-	-	-	2.89	**2.25**	9.72

At first energy based voice activity detection is performed on each utterance. Then, feature extraction is performed using a 25ms hamming window with 50% overlap (12.5ms). Voicebox MATLAB toolbox [1] is employed to extract MFCC features. To obtain PLP

[1] Available online at: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.zip

features, we used RASTAMAT MATLAB toolbox which is available online[1]. SWLP features, which we have used, are briefly described in [21], and there are MATLAB codes available online[2], to extract these features. To extract LPCC features, we used msf_lpcc.m MATLAB implementation[3]. We choose a feature vector dimension of 14 for all four features.

We used a 2048 Gaussian mixture model to create a universal background model on the training part of NIST 2004 speaker recognition evaluation corpus. MSR toolbox is used to extract all GMM models, and i-vectors [22]. GMM_UBM_KL and ivector_PLDA are implemented using MSR toolbox and voicebox. The i-vector dimensionality was 400, which is reduced to 200 using PLDA. An important matter in score fusion is that, scores from different classifiers may vary significantly, as a result of using different feature vectors and classification methods. Using results of [2] we used z-cal(clipped) as pre-calibration method.

To evaluate base classifiers and fusion method we considered EER, minDCF and minCLLR using BOSARIS MATLAB toolkit [23].

4.3 Results

In this section we consider three methods to fuse individual scores. In the first method we use a weighted logistic regression cost, C_{wlr}, regulated by E-net ($\alpha = 0.1$) [2]. In the second method we replace the regularization term with our proposed sample specific term, and perform ensemble selection using the clarity index (IBSparse1). In this method optimization is performed on each test sample. In the last experiment we calculated weight vectors for all possible combinations of the ensemble set (IBSparse2). We empirically found that best results can be obtained when the size of the ensemble is limited between 4 and 8 classifiers. In this situation, most suitable classifiers are selected based on the test sample and the optimization of weights is not performed in the test stage.

Table 1 shows that different classifiers have instance based behaviors. For example ivector-PLDA which uses PLP features, has the best EER for the development set, while this is not suitable for the evaluation set. The next evidence is the whole performance of GMM-SVM-KL in comparison to which is good with respect to other classification methods and is worse for the evaluation set. Comparing the performance of GMM-SVM-KL using LPCC features and ivector-PLDA using MFCC features supports the same idea.

As an example of performance improvement, we observed, sparse classifier fusion [2] results in the score of 5.1484 for the verification of the utterance '*xalm.sph*' which belongs to NIST SRE 2004, and class 1 (the first model in the database). Because this is a target score, it is better to be higher. The clarity index for this trial is as follows:

[0.845,0.555,0.825,0.66,0.355,0.64,0.935,0.53,0.72,0.87,0.98,0.875]

The proposed method chooses the 6 most confident classifiers which are: 1st, 3rd, 7th, 10th, 11th and 12th classifiers (of Table 1). The Fusion of scores of these classifiers results in a fused score of 7.2706.

To use the clarity index in ensemble set selection, a threshold value should be used. Values higher than the threshold are considered as confident classifiers and lower values are considered as belonging to unconfident classifiers. We obtained the threshold value from the development set and used it in the evaluation set for classifier ensemble selection. A comparison of three speaker verification fusion systems is shown in Figure 3. This curve is obtained using MSR MATLAB toolbox. It is clearly observed that the proposed method 1 shows the best results in almost all parts of the plot, with the cost of optimization of weights in the test phase.

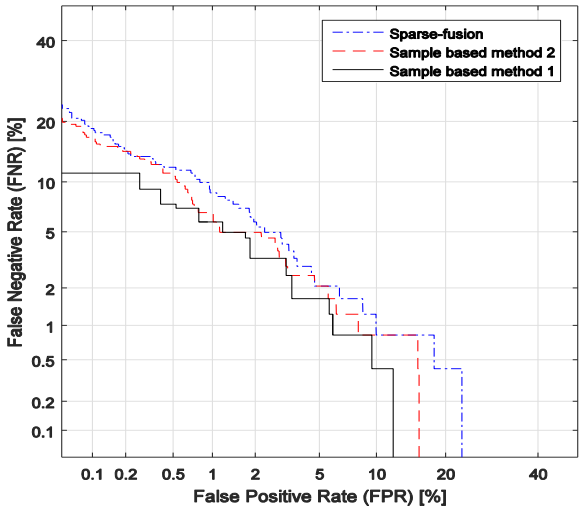

Fig. 3. DET plot of three speaker verification fusion systems (plotted using MSR MATLAB toolbox)

4.4 Correlation of Classifiers

Diversity of classifiers is an important issue in ensemble classification. A More diverse set of classifiers increases the chance of taking more aspects of the classification in to account. Correlation is the opposite point of diversity. One can use one of the two highly correlated classifiers without significant reduction in performance. In the case of our experiment we indirectly take the correlation into account. As it is mentioned in subsection 3.2.1, if the score of a classifier is less than all non-target scores, the classifier confidently tells that test sample does not belong to the claimed class and if the value is greater than all the target scores the classifier confidently tells that test sample does belong to the claimed class. In both the situations CL value is 1. Therefor higher values of CL belong to confident classifiers and lower values belong to unconfident ones. By choosing a proper threshold value of for the clarity we omit very unconfident classifiers. Therefore the remaining classifiers are assumed to be confident enough to

[1] Available online at: http://labrosa.ee.columbia.edu/matlab/rastamat
[2] Available online at: http://users.spa.aalto.fi/jpohjala/xlp/
[3] Available online at: https://github.com/jameslyons/matlab_speech_features/archive/master.zip

participate in the ensemble. A question that is raised here is what happens if some of the classifiers are highly correlated. For example, if seven classifiers are in the ensemble and four of them are highly correlated, it means they exploit the same speech characteristics and lower the effectiveness of other uncorrelated classifiers. If the weight of every classifier is fixed for all test samples, this effect reduces the performance of the ensemble, due to the fact that classifiers have different correlations for different samples. This issue remains while weight learning is performed in the development phase, including our sample specific fusion method 2. But when weights are learned in the test phase, even if the mentioned four classifiers are in the ensemble set, and exploit the exact same characteristics of the utterance, the weight learning algorithm gives them the most efficient weights.

5. Conclusions

We introduced a sample specific classifier fusion for speaker verification, which selects an adaptive number of best classifiers and determines sample specific fusion weights for each selected classifier. The method implements a group of well-known base classifiers for speaker verification, and ranks them using information obtained from labeled samples and individual unlabeled samples. The weight learning process uses logistic regression and the optimization problem is constrained with a sample specific term.

Extensive experiments on unconditioned, large variant NIST 2004 demonstrated the effectiveness of the proposed method. It would be interesting to perform experiments about the weight of constraint (λ) and the timing of the optimization formula.

References

[1] X. Zhou, A. S. d'Avila Garcez, H. Ali, S. N. Tran, and K. Iqbal, "Unimodal late fusion for NIST i-vector challenge on speaker detection," Electron. Lett., vol. 50, no. 15, pp. 1098–1100, 2014.

[2] H. L. Ville Hautamä ki, Tomi Kinnunen, Filip Sedlák, Kong Ail Lee, Bin Ma, "Sparse Classifier Fusion for Speaker Verification," IEEE Trans. Audio, Speech Lang. Process., vol. 21, no. 8, pp. 1622–1631, 2013.

[3] Z. Lei, Y. Yang, and Z. Wu, "Ensemble of support vector machine for text-independent speaker recognition," IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 6, no. 5, pp. 163–167, 2006.

[4] A. Kumar and B. Raj, "Unsupervised Fusion Weight Learning in Multiple Classifier Systems," arXiv:1502.01823, Feb. 2015.

[5] K. Lai, D. Liu, S. Chang, and M. Chen, "Learning Sample Specific Weights for Late Fusion," Image Process. IEEE Trans., vol. 24, no. 9, pp. 2772–2783, 2015.

[6] R. Saeidi, J. Pohjalainen, T. Kinnunen, and P. Alku, "Temporally weighted linear prediction features for tackling additive noise in speaker verification," IEEE Signal Process. Lett., vol. 17, no. 6, pp. 599–602, 2010.

[7] F.I. Cabeceran, "Fusing prosodic and acoustic information for robust speaker recognition," Ph.D dissertation, TALP Research Center, Speech Processing Group, Universitat Politècnica de Catalunya Barcelona, July 2008.

[8] M. J. Alam, T. Kinnunen, P. Kenny, P. Ouellet, and D. O'Shaughnessy, "Multitaper MFCC and PLP features for speaker verification using i-vectors," Speech Commun., vol. 55, no. 2, pp. 237–251, Feb. 2013.

[9] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation," 2006 IEEE Int. Conf. Acoust. Speech Signal Process. Proc., vol. 1, no. 2, pp. 1–3, 2006.

[10] C. You, K.-A. Lee, and H. Li, "GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition," IEEE Trans. Audio, Speech Lang. Process., vol. 18, no. 6, pp. 1300–1312, 2010.

[11] P. Mat, M. Kara, and P. Kenny, "full-covariance ubm and heavy-tailed plda in i-vector speaker verification," Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, Prague, Czech Republic, 22 May - 27 May 2011 pp. 4516–4519, 2011.

[12] A. Solomonoff, W. M. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Philadelphia PA, USA, 2005, pp. 629–632.

[13] P. Emerson, Designing an All-Inclusive Democracy. Springer, 2007.

[14] W. M. Campbell, D. E. Sturim, W. Shen, D. A. Reynolds, and J. Navrátil, "The MIT-LL/IBM 2006 speaker recognition system: High-performance reduced-complexity recognition," in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Honolulu, Hawaii, USA, 15-20 April 2007.

[15] S. Chernbumroong, S. Cang, and H. Yu, "Genetic Algorithm-based Classifiers fusion for multi-sensor activity recognition of elderly people," IEEE J. Biomed. Heal. informatics, vol. 19, no. 1, pp. 282 – 289, 2014.

[16] C. M. Bishop, Pattern recogintion and machine learning (Information Science and Statistics). Springer, 2007.

[17] S. Pigeon, P. Druyts, and P. Verlinde, "Applying logistic regression to the fusion of the NIST'99 1_speaker submissions," Digit. Signal Process., vol. 10, no. 1–3, pp. 237–248, 2000.

[18] M. Schmidt, G. Fung, and R. Rosales, "Fast Optimization Methods for L1 Regularization: A Comparative Study and Two New Approaches," Lect. Notes Comput. Sci., vol. 4701, pp. 286–297, 2007.

[19] N. Brümmer, "Measuring, refining and calibrating speaker and language information extracted from speech," Ph.D dissertation, Faculty of Engineering, University of Stellenbosch, 2010.

[20] S. Wang and X. Yao, "Relationships between diversity of classification ensembles and single-class performance measures," IEEE Trans. Knowl. Data Eng., vol. 25, no. 1, pp. 206–219, 2013.

[21] C. Magi, J. Pohjalainen, T. Bäckström, and P. Alku, "Stabilised weighted linear prediction," Speech Commun., vol. 51, no. 5, pp. 401–411, 2009.
[22] S. O. Sadjadi, M. Slaney, and L. Heck, "MSR Identity Toolbox v1. 0: A matlab toolbox for speaker-recognition research," Speech Lang. Process. Tech. Comm. Newsl., 2013.
[23] N. Brümmer and E. de Villiers, "Bosaris toolkit [software package]," 2011. [Online]. Available: https://sites.google.com/site/bosaristoolkit/.

Mohammad Hasheminejad received the B.Sc. degree in Bio-electrical engineering from University of Isfehan, Isfehan, Iran, in 2003. He received the M.Sc. degree in communication engineering from Maleke ashtar university of technology, Tehran, Iran, in 2008. He is currently Ph.D student in Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. His area research interests include Image Processing and retrieval, Pattern recognition, Digital Signal Processing and Sparse representation. His email address is: mhashemi@birjand.ac.ir.

Hassan Farsi received the B.Sc. and M.Sc. degrees from Sharif University of Technology, Tehran, Iran, in 1992 and 1995, respectively. Since 2000, he started his Ph.D in the Centre of Communications Systems Research (CCSR), University of Surrey, Guildford, UK, and received the Ph.D degree in 2004. He is interested in speech, image and video processing on wireless communications. Now, he works as associate professor in communication engineering in department of Electrical and Computer Eng., university of Birjand, Birjand, IRAN. His Email is: hfarsi@birjand.ac.ir.