**EURASIP Journal on Audio, Speech, and Music Processing**

SpringerOpen



# Homogenous ensemble phonotactic language recognition based on SVM supervector reconstruction

*EURASIP Journal on Audio, Speech, and Music Processing* 2014, **2014**:42 doi:10.1186/s13636-014-0042-5

Wei-Wei Liu (liu-ww10@hotmail.com)
Wei-Qiang Zhang (wqzhang@tsinghua.edu.cn)
Michael T Johnson (michael.johnson@marquette.edu)
Jia Liu (liuj@tsinghua.edu.cn)

Published online: 24 December 2014

**ISSN** 1687-4722

**Article type** Research

**Submission date** 14 May 2014

**Acceptance date** 28 October 2014

**Article URL** http://asmp.eurasipjournals.com/content/2014/1/42

# Homogenous ensemble phonotactic language recognition based on SVM supervector reconstruction

Wei-Wei Liu[1,2]
Email: liu-ww10@hotmail.com

Wei-Qiang Zhang[1*]
*Corresponding author
Email: wqzhang@tsinghua.edu.cn

Michael T Johnson[3]
Email: michael.johnson@marquette.edu

Jia Liu[1]
Email: liuj@tsinghua.edu.cn

[1]Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

[2]General Communication Station, General Logistics Department, Beijing 100036, China

[3]Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI 53233, USA

## Abstract

Currently, acoustic spoken language recognition (SLR) and phonotactic SLR systems are widely used language recognition systems. To achieve better performance, researchers combine multiple subsystems with the results often much better than a single SLR system. Phonotactic SLR subsystems may vary in the acoustic features vectors or include multiple language-specific phone recognizers and different acoustic models. These methods achieve good performance but usually compute at high computational cost. In this paper, a new diversification for phonotactic language recognition systems is proposed using vector space models by support vector machine (SVM) supervector reconstruction (SSR). In this architecture, the subsystems share the same feature extraction, decoding, and N-gram counting preprocessing steps, but model in a different vector space by using the SSR algorithm without significant additional computation. We term this a homogeneous ensemble phonotactic language recognition (HEPLR) system. The system integrates three different SVM supervector reconstruction algorithms, including relative SVM supervector reconstruction, functional SVM supervector reconstruction, and perturbing SVM supervector reconstruction. All of the algorithms are incorporated using a linear discriminant analysis-maximum mutual information (LDA-MMI) backend for improving language recognition evaluation (LRE) accuracy. Evaluated on the National Institute of Standards and Technology (NIST) LRE 2009 task, the proposed HEPLR system achieves better performance than a baseline phone recognition-vector space modeling (PR-VSM) system with minimal extra computational cost. The performance of the HEPLR system yields 1.39%, 3.63%, and 14.79% equal error rate (EER), representing 6.06%, 10.15%, and 10.53% relative improvements over the baseline system, respectively, for the 30-, 10-, and 3-s test conditions.

**Keywords**

Phonotactic language recognition; Support vector machine (SVM) supervector reconstruction; Phone recognition-vector space modeling (PR-VSM)

## 1 Introduction

Spoken language recognition (SLR) refers to the task of automatic determination of language identity. It is estimated that there are about 6,000 spoken languages in the world [1]. An increasing number of multilingual speech processing applications require spoken language recognition as a frontend, with the result that SLR continues to grow in importance. Spoken language recognition is an enabling technology for a wide range of intelligence and security applications for information distillation, such as spoken document retrieval, multilingual speech recognition, and spoken language translation [2].

Language cues can be categorized according to their level of knowledge abstraction as acoustic (spectrum, phone inventory), prosodic (duration, pitch, intonation), phonotactic (sequence of sounds), lexical (vocabulary, morphology), and syntax (phrases, grammar) [3,4]. Language recognition systems are usually identified by the features they employ, e.g., acoustic systems, phonotactic systems, prosodic systems, and lexical systems. Currently, acoustic language recognition (LR) systems [5] and phonotactic LR systems [3] are both widely used.

Generally, the performance of SLR systems can be improved in two ways: (1) longitudinally, through the development of new techniques to perform the SLR tasks more precisely, e.g., i-vector [6-8], JFA [9], discriminative training [10] methods, $N$-gram modeling methods [3], and support vector machines (SVMs) [11]; (2) transversely, by adding variety to the SLR subsystems, which extracts and integrates more information from the utterances.

State-of-the-art language recognition systems fuse multiple subsystems in parallel via a post-processing backend [12].

In the National Institute of Standards and Technology (NIST) language recognition evaluation (LRE) tasks, teams from all over the world compete to build the best SLR system and have shown that better results can be obtained by combining more subsystems, creating larger and larger SLR systems. In NIST LRE 2011, all submitted language recognition systems were stacked ensembles of at least five language recognition subsystems [13-15]. Much effort goes into trying different variations of subsystems. Generally, the phonotactic LR subsystems can be varied in three ways: (a) extracting various acoustic features to provide feature diversification, for example, Mel-Frequency Cepstral Coefficients (MFCC) [16], Perceptual Linear Predictive (PLP) [17], and Temporal Patterns Neural Network (TRAPs/NN) [18]; (b) training phone recognizers on multiple language-specific speech data to provide phonetic diversification [3], e.g., the Russian, Hungarian, Czech, and English phone recognizers developed by Brno University of Technology (BUT) [18] or universal phone recognizer (UPR) [19]; and (c) training phone recognizers on the same language-specific speech data but using different acoustic models to provide acoustic diversification [20], such as the Artificial Neural Network-Hidden Markov Model (ANN-HMM) [21], Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) [22], and Deep Neural Network-Hidden Markov Model (DNN-HMM) [23]. Certainly, in phonotactic language recognition systems, the subsystems must undergo different process of feature extracting, decoding, N-gram counting, and vector space modeling, which means an added computational cost of $N$ times than single subsystem, where $N$ is the number of the subsystems.

This paper demonstrates an architecture to provide a new diversification for phonotactic language recognition systems. The underlying motivation of these algorithms is to provide richer language

identifying information without significant additional computation. The subsystems are verified using SVM supervector reconstruction (SSR) algorithms to provide vector space modeling diversification. In this architecture, the subsystems share the same preprocessing of feature extracting, decoding, and expected counting, but models in different vector space, so we call it homogeneous ensemble phonotactic language recognition (HEPLR) system. The HEPLR subsystems increase the variety of the SVM supervector, decrease the computational cost, and improve the SLR accuracy. There are many SVM supervector reconstruction algorithms such as recurrent neuron network (RNN) SVM supervector reconstruction [24]. In this paper, we present three SVM supervector reconstruction algorithms including relative SVM reconstruction [25], functional SVM reconstruction, and perturbative SVM reconstruction.

The remainder of the paper is organized as follows. Section 2 presents the lattice-based phonotactic language recognition system used as a baseline in this paper. Section 3 describes the proposed approaches, includes relative, functional, and perturbative SVM supervector reconstruction. Section 4 demonstrates the architecture for the homogeneous ensemble phonotactic language recognition system. The experimental setup to evaluate our proposed method is described in Section 5. Results obtained in language recognition experiments on the NIST LRE 2009 database are presented and discussed in Section 6. Finally, conclusions and future work are outlined in Section 7.

## 2 Baseline phonotactic SLR system

In this work, the phone recognition-vector space modeling (PR-VSM) [26] phonotactic language recognition system is employed as a baseline system. The motivation behind phonotactic language recognition approach is the belief that a spoken language can be characterized by special probabilities of its lexical-phonological constraints. An $N$-gram vector space model (VSM) is built for the language recognition task using phone transcriptions, which is a stochastic model describing the probabilities of phoneme strings. In the PR-VSM system, each $N$-gram VSM of produces a likelihood score by SVM classifier to a given utterance. The languages used for training the phone recognizers need not be the same with any of those recognized.

The traditional PR-VSM language recognition system works by mapping the input utterances from data space $\mathcal{X}$ into a high-dimensional feature space $\mathcal{F}$: $\Phi : \mathcal{X} \rightarrow \mathcal{F}$ and then building linear machines in the feature space to find a maximal margin separation. The vectors built in the high-dimensional feature space are SVM supervectors, which consist of $N$-gram counts of features representing the phonotactics of an input speech wave sample.

In PR-VSM systems, an utterance $x$ can be mapped to the high-dimensional feature space as follows:

$$\mathbf{\Phi} : x \rightarrow \varphi(x), \tag{1}$$

where $\varphi(x)$ is the SVM supervector computed as

$$\varphi(x) = [p(d_1|\ell_x), p(d_2|\ell_x), \ldots, p(d_F|\ell_x)], \tag{2}$$

where $\ell_x$ is the lattice produced from data $x$ by a phone recognizer, $d_i$ is the $N$-gram phoneme string [27] $d_i = s_i \ldots s_{i+n-1}$ $(n = N)$, and $F$ is the dimension of the SVM supervector. $p(d_i|\ell_x)$ is the probability of the phoneme sequence $d_i$ in the lattice, which is computed as

$$p(d_i|\ell_x) = \frac{c(d_i|\ell_x)}{\sum_i c(d_i|\ell_x)},$$

where $c(d_i|\ell_x)$ denotes the $N$-gram occurrence of $d_i$ given the lattice $\ell_x$. This is calculated over all

possible hypotheses in the lattice as follows [27]:

$$c(s_i...s_{i+N-1}|\ell) = \sum_{S\in\ell} p(S|\ell)c(s_i...s_{i+N-1}|S)$$
$$= \sum_{s_i...s_{i+N-1}\in\ell} [\alpha(s_i)\beta(s_{i+N-1}) \prod_{j=i}^{i+N-1} \xi(s_j)],$$

where $p(S|\ell)$ denotes the probability of the sequence $S$ in the lattice $\ell$, $\alpha(s_i)$ and $\beta(s_{i+N-1})$ are the forward probability of the starting node and the backward probability of the ending node in the $N$-gram $s_i...s_{i+N-1}$, respectively. $\xi(s_j)$ is the posterior probability of the edge phoneme $s_j$.

The SVM supervector $\varphi(x)$ is sent to the SVM classifier and a decision is made based on the most likely hypothesis score. In a PR-VSM system, the decision is made based on the SVM output score using

$$f(\varphi(x)) = \sum\nolimits_l \alpha_l K_{\mathrm{TFLLR}}(\varphi(x), \varphi(x_l)) + d, \tag{3}$$

where $\varphi(x_l)$ are support vectors obtained from training set using the Mercer condition. The term frequency log-likelihood ratio (TFLLR) kernel $K_{\mathrm{TFLLR}}$ is computed as [28]:

$$K_{\mathrm{TFLLR}}(\varphi(x_i), \varphi(x_j)) = \sum_{q=1}^{F} \frac{p(d_q|\ell_{x_i})}{\sqrt{p(d_q|\ell_{all})}} \frac{p(d_q|\ell_{x_j})}{\sqrt{p(d_q|\ell_{all})}}, \tag{4}$$

where $p(d_i|\ell_{all})$ is calculated from the observed probability of $d_i$ across all lattices. In this paper, the training stage is always carried out with a one-versus-rest strategy between the positive set (the samples in the target language) and negative set (all other samples).

## 3 SVM supervector reconstruction algorithms

The motivation behind SVM supervector reconstruction is to provide vector space modeling diversification to improve the performance of the overall language recognition system. In the language recognition system employed in this paper, we focus on how a change in the input to the SVM affects the output.

Given an SVM supervector $\varphi(x)$, we define a function $\phi_{\mathrm{SSR}}$ which operates on $\varphi(x)$:

$$\mathbf{\Phi}_{\mathrm{SSR}} : x \rightarrow \phi_{\mathrm{SSR}}(\varphi(x)). \tag{5}$$

We are interested in understanding how $\phi_{\mathrm{SSR}}(\varphi(x))$ affects the behavior of the output scores of the SVM. The goal is to define the relationship between $\varphi(x)$ and $\phi_{\mathrm{SSR}}(\varphi(x))$ to enhance the variety of the supervector input.

Selecting SVM supervector reconstruction methods is an open question, so here we propose some typical methods to the implementation. In this section, three SVM supervector reconstruction methods are proposed: relative SVM supervector reconstruction, functional SVM supervector reconstruction, and perturbative SVM supervector reconstruction. Relative SVM supervector reconstruction has been presented in [25], while functional and perturbative reconstructions are new methods. Relative reconstruction is a linear reconstruction, while functional and perturbative reconstructions are non-linear ones.

### 3.1 Relative SVM supervector reconstruction

The relative SVM supervector method uses a relative feature approach. Relative features in contrast to absolute features, which represent directly calculable information, are defined by the relationship between an utterance and a set of selected datum utterances. We have presented the concept of relative features in [25].

Calculating relative features requires a relationship measurement, such as distance on similarity. By selecting a proper relationship measurement, the identifiable characteristics can be strengthened and nuisance attributes of the utterance can be discarded. Unlike absolute features, relative features make utterances more convenient to classify by showing the relationship between the utterances and the datum database directly.

Here, we introduce a relative SVM supervector reconstruction defined using the similarity between the utterance SVM supervectors. The widely used kernel methods offer efficient similarity measurements between two SVM supervectors. In this paper, the empirical kernel [29] is introduced into language recognition and a relativized SVM supervector developed. Kernel methods have been used for face recognition [30] and handwritten digit recognition [31] and achieved higher robustness to noise [30]. Using the SVM supervectors that are already built into a language recognition system, we can easily compose a new relativized SVM supervector with only a small increase in computation.

The architecture of the relative SVM supervector reconstruction subsystem is shown in Figure 1. To construct the SVM supervector relativization map, a database $\boldsymbol{s} = [s_1, s_2, ...s_m]$ containing $m$ utterances is used as the datum mark of similarity. The datum database is stochastically selected from some corpus, whose language need not be the same with the target language. $\boldsymbol{s}$ is mapped into vector space:

$$\boldsymbol{s} \to \varphi(\boldsymbol{s}) = [\varphi(s_1), \varphi(s_2), ..., \varphi(s_m)]. \tag{6}$$

The vector relativizational (VR) kernel between two supervectors $\varphi(x_i)$ and $\varphi(x_j)$ is

$$\begin{aligned} K_{\mathrm{VR}}(\varphi(x_i), \varphi(x_j)) &=< \varphi(x_i), \varphi(x_j) > \\ &= \sum_{q=1}^{F} \frac{p(d_q|\ell_{x_i})}{\sqrt{p(d_q|\ell_S)}} \frac{p(d_q|\ell_{x_j})}{\sqrt{p(d_q|\ell_S)}}. \end{aligned} \tag{7}$$

**Figure 1 Architecture of relative SVM supervector reconstruction subsystem.**

The VR kernel is similar to the TFLLR kernel, but normalized by the observed probability across all lattices of the datum dataset $p(d_i|\ell_s)$. This kernel reflects the degree of similarity between two supervectors.

The utterance $x$ is mapped from the input data space $\mathcal{X}$ to a relativized $m$-dimensional Euclidean space $\mathcal{R}^m$: $\Phi_{\mathrm{REL}} : \mathcal{X} \to \mathcal{R}^m$ as follows:

$$\begin{aligned} &\boldsymbol{\Phi_{\mathrm{REL}}} : x \to \varphi_{\mathrm{REL}}(x) \\ &= K_{\mathrm{VR}}(\varphi(x), \varphi(\boldsymbol{s})) =< \varphi(x), \varphi(\boldsymbol{s}) > \\ &= [K_{\mathrm{VR}}(\varphi(x), \varphi(s_1)), ..., K_{\mathrm{VR}}(\varphi(x), \varphi(s_m))]. \end{aligned}$$

In general, $K_{\mathrm{VR}}(\varphi(x), \varphi(\boldsymbol{S}))$ defines a space in which each dimension corresponds to the similarities to a prototype. Thus, $K_{\mathrm{VR}}(\cdot, \varphi(\boldsymbol{S}))$ can be viewed as a mapping onto an $m$-dimensional relativized vector space.

The SVM output score is computed as

$$f'(\varphi_{\mathrm{REL}}(x)) = \sum\nolimits_{l'} \alpha_{l'} K'(\varphi_{\mathrm{REL}}(x), \varphi_{\mathrm{REL}}(x_{l'})) + d', \tag{8}$$

where $\varphi_{\mathrm{REL}}(x_{l'})$ are support vectors obtained from the training set using the Mercer condition. Selecting a radial basis function (RBF) kernel, $K'_{\mathrm{RBF}}$ is computed as

$$K'_{\mathrm{RBF}}(\varphi_{\mathrm{REL}}(x_i), \varphi_{\mathrm{REL}}(x_j)) = \exp(-\frac{|\varphi_{\mathrm{REL}}(x_i) - \varphi_{\mathrm{REL}}(x_j)|^2}{D_{\mathrm{RF}}}), \tag{9}$$

where $D_{\mathrm{RF}}$ is the dimension of the relativized SVM supervector. Selecting a TFLLR kernel, $K'_{\mathrm{TFLLR}}$ is computed as

$$K'_{\mathrm{TFLLR}}(\varphi_{\mathrm{REL}}(x_i), \varphi_{\mathrm{REL}}(x_j)) = \sum_{q=1}^{m} \frac{K_{\mathrm{VR}}(\varphi(x_i), \varphi(s_q)) K_{\mathrm{VR}}(\varphi(x_j), \varphi(s_q))}{K_{\mathrm{VR}}(\varphi(x_{\mathrm{all}}), \varphi(s_q))}. \tag{10}$$

### 3.2 Functional SVM supervector reconstruction

In actual test conditions, the training and test data are variable in speakers, background noise, and channel conditions. To achieve higher robustness to variable test conditions, the widely used kernel methods offer efficient similarity measurements between two SVM supervectors in PR-VSM system [30]. The geometrical structure of the SVM vector space is completely determined by the kernel, so the selection of the kernel has a crucial impact on the performance of the language recognition systems. The functional SVM supervector reconstruction method defines a mixture between the functional and the original kernels, which can offer the robust discriminative information of the data and get robust language model.

But how to select a proper function is an open problem. There are many functions that can be used to the reconstruction, while not every function is available for the reconstruction that can reduce the equal error rate (EER). What we need to do is to find out what kind of functions can be used in feature reconstruction. The functions need to satisfy the following conditions: (1) monotonic and (2) can make the identifiable characteristics strengthened and nuisance attributes of the utterance discarded. The proposed functional SVM supervector reconstruction method does not rely on prior knowledge to select the functional to reconstruct the supervector. A development database is used for cross validation to select the function. So, here, three functions selected to be used in this paper include

(a)

$$\varphi_{\mathrm{FUN}}(p(d_i|\ell_x)) = \sin(p(d_i|\ell_x)) + \cos(p(d_i|\ell_x)), \tag{11}$$

(b)

$$\varphi_{\mathrm{FUN}}(p(d_i|\ell_x)) = p(d_i|\ell_x) + (p(d_i|\ell_x))^2, \tag{12}$$

(c)

$$\varphi_{\mathrm{FUN}}(p(d_i|\ell_x)) = p(d_i|\ell_x) - (p(d_i|\ell_x))^2 + (p(d_i|\ell_x))^3. \tag{13}$$

The utterance $x$ is mapped onto a functionalized vector space:

$$\mathbf{\Phi}_{\mathrm{FUN}} : x \to \varphi_{\mathrm{FUN}}(x) = [\varphi_{\mathrm{FUN}}(p(d_1|\ell_x)), \varphi_{\mathrm{FUN}}(p(d_2|\ell_x)), ..., \varphi_{\mathrm{FUN}}(p(d_F|\ell_x))]. \tag{14}$$

The three functions are all monotonic in the range of amplitude of the SVM supervector. Selecting a

TFLLR kernel, $K'_{\mathrm{TFLLR}}$ is computed as

$$K'_{\mathrm{TFLLR}}(\varphi_{\mathrm{FUN}}(x_i), \varphi_{\mathrm{FUN}}(x_j)) = \sum_{q=1}^{F} \frac{\varphi_{\mathrm{FUN}}(p(d_q|\ell_{x_i}))\varphi_{\mathrm{FUN}}(p(d_q|\ell_{x_j}))}{\varphi_{\mathrm{FUN}}(p(d_q|\ell_{all}))}. \quad (15)$$

The SVM output score is computed as

$$f'(\varphi_{\mathrm{FUN}}(x)) = \sum\nolimits_{l'} \alpha_{l'} K'(\varphi_{\mathrm{FUN}}(x), \varphi_{\mathrm{FUN}}(x_{l'})) + d'. \quad (16)$$

### 3.3 Perturbational SVM supervector reconstruction

For spoken language recognition, the first and most essential step is to tokenize the running speech into sound units or lattices using a phone recognizer. The phoneme error rate is around 40% to 60% [32] when tokenizing an utterance. The decoding errors are deletion, insertion, and substitution errors, which are expressed as some discrete 'noise' when mapped to the high-dimensional SVM supervector space (shown in Figure 2). So, here, we introduce a perturbational denoising method for the SVM supervector. Given a supervector $\varphi(x)$ and some perturbation operator on $\varphi(x)$, we are interested in understanding how a small perturbation added to the supervector affects the behavior of the SVM [33]. This relationship can be represented using a mapping onto a perturbational vector space.

**Figure 2 Effections of decoding errors on SVM supervector.**

There are three purposes of proposing perturbational SVM supervector reconstruction method: first, adding perturbational noise to reduce the impact of noise in the SVM supervector introduced by the decoding errors; second, generating a more robust language model to provide input variety to the SVM classifier; and third, highlighting the most discriminative information of the SVM supervector and drowning the non-discriminative information into the perturbation (shown in Figure 3).

**Figure 3 SVM supervector of an utterance (a) before and (b) after perturbation.**

To accomplish the above goals, the type and strength of the perturbation must be selected carefully. How to define a proper perturbation is an open problem. There are a wide variety of perturbations,which can be categorized into multiple ways, including (1) global perturbation and local perturbation, (2) stochastic perturbation and constant perturbation according to the amplitude, (3) absolute perturbation and relative perturbation according to the relationship between the SVM supervector and the perturbation, and (4) addictive perturbation and multiplicative perturbation.

For feature supervectors in vector space, the perturbations are always discrete, maybe random in a certain range or change with the amplitude of the expected value of the supervector. So, we consider both deterministic perturbation $\boldsymbol{\delta} = w^* E_{p(d|\ell_x)}$ and stochastic perturbation $\boldsymbol{\delta} = w^* \mathrm{uniform}[0, E_{p(d|\ell_x)}]$, where $E_{p(d|\ell_x)}$ is the mean of the SVM supervector, $\mathrm{uniform}[0, E_{p(d|\ell_x)}]$ is the uniform distribution between 0 and $E_{p(d|\ell_x)}$, and $w^*$ is the perturbation weight. More details of the perturbation methods are discussed below.

#### *3.3.1 Perturbational approach 1 (deterministic additive perturbation)*

$$\boldsymbol{\delta} = w^* E_{p(d|\ell_x)}, \quad \varphi_{\mathrm{PER}}(p(d_i|\ell_x)) = p(d_i|\ell_x) + \boldsymbol{\delta}. \quad (17)$$

This kind of perturbation represents the assumption that the expected count of every phoneme sequence is perturbed by an equivalent additive amount.

#### *3.3.2 Perturbational approach 2 (stochastic additive perturbation)*

$$\boldsymbol{\delta} = w^* \text{random}[0, E_{p(d|\ell_x)}], \quad \varphi_{\text{PER}}(p(d_i|\ell_x)) = p(d_i|\ell_x) + \boldsymbol{\delta}. \tag{18}$$

This perturbation represents the assumption that the expected count of every phoneme sequence is perturbed by an amount proportional to the frequency of the phoneme sequences.

#### *3.3.3 Perturbational approach 3 (deterministic multiplicative perturbation)*

$$\boldsymbol{\delta} = w^* E_{p(d|\ell_x)}, \quad \varphi_{\text{PER}}(p(d_i|\ell_x)) = p(d_i|\ell_x)\boldsymbol{\delta}. \tag{19}$$

This kind of perturbation represents the assumption that the expected count of every phoneme sequence perturbed by an equivalent multiple amount.

#### *3.3.4 Perturbational approach 4 (stochastic multiplicative perturbation)*

$$\boldsymbol{\delta} = w^* \text{random}[0, E_{p(d|\ell_x)}], \quad \varphi_{\text{PER}}(p(d_i|\ell_x)) = p(d_i|\ell_x)\boldsymbol{\delta}. \tag{20}$$

This perturbation represents the assumption that the expected count of every phoneme sequence is perturbed by a proportional to the frequency of the phoneme sequences.

From above, it can be seen that methods 1 and 2 implement absolute perturbation, and methods 3 and 4 implement relative perturbation. All are global perturbation algorithms, operating across the entire vector space. We can also investigate local perturbation using these same approaches. Local perturbation is more flexible and realistic for the noises would have effect on part of the expected counting. The proposed methods also do not rely on prior knowledge to put noising into the supervector; we use development database for cross validation to select a better perturbation.

The utterance $x$ is mapped onto a perturbational vector space:

$$\boldsymbol{\Phi}_{\text{PER}} : x \rightarrow \varphi_{\text{PER}}(x) = [\varphi_{\text{PER}}(p(d_1|\ell_x)), \varphi_{\text{PER}}(p(d_2|\ell_x)), ..., \varphi_{\text{PER}}(p(d_F|\ell_x))]. \tag{21}$$

where $\varphi_{\text{PER}}(x)$ is a perturbation of $\varphi(x)$. Selecting a TFLLR kernel, $K'_{\text{TFLLR}}$ is computed as

$$K'_{\text{TFLLR}}(\varphi_{\text{PER}}(x_i), \varphi_{\text{PER}}(x_j)) = \sum_{q=1}^{F} \frac{\varphi_{\text{PER}}(p(d_q|\ell_{x_i}))\varphi_{\text{PER}}(p(d_q|\ell_{x_j}))}{\varphi_{\text{PER}}(p(d_q|\ell_{all}))}. \tag{22}$$

The SVM output score is computed as

$$f'(\varphi_{\text{PER}}(x)) = \sum\nolimits_{l'} \alpha_{l'} K'(\varphi_{\text{PER}}(x), \varphi_{\text{PER}}(x_{l'})) + d'. \tag{23}$$

## 4 Homogeneous ensemble language recognition system

The architecture of the HEPLR system is shown in Figure 4. All the SVM supervectors are reconstructed into the corresponding vector space and fused at the vector level for training and testing. In this paper, we use the classical method of vector fusion, which is to group several sets of reconstructed SVM supervectors into a large composite supervector [34]. Suppose $\varphi_{\mathrm{REL}_{N_{1_1}}}(x)$, ..., $\varphi_{\mathrm{REL}_{N_{1'_m}}}(x)$, $\varphi_{\mathrm{FUN}_{N_{2_1}}}(x)$, ..., $\varphi_{\mathrm{FUN}_{N_{2'_m}}}(x)$, and $\varphi_{\mathrm{PER}_{N_{3_1}}}(x)$, ..., $\varphi_{\mathrm{PER}_{N_{3'_m}}}(x)$ are the reconstructed SVM supervectors for an input utterance $x$. The concatenated SVM supervectors can be represented as $\varphi_{\mathrm{REL}}(x)$, $\varphi_{\mathrm{FUN}}(x)$, and $\varphi_{\mathrm{PER}}(x)$, respectively. Denoting each SVM supervector as $d_m$-dimensional, the concatenated SVM supervectors are $(d_1 + ... + d'_m)$-dimensional. The concatenated SVM supervectors are defined by

$$\varphi_{\mathrm{REL}}(x) = [w_{N_{1_1}}\varphi_{\mathrm{REL}_{N_{1_1}}}(x), ..., w_{N_{1'_m}}\varphi_{\mathrm{REL}_{N_{1'_m}}}(x)], \tag{24}$$

$$\varphi_{\mathrm{FUN}}(x) = [w_{N_{2_1}}\varphi_{\mathrm{FUN}_{N_{2_1}}}(x), ..., w_{N_{2'_m}}\varphi_{\mathrm{FUN}_{N_{2'_m}}}(x)], \tag{25}$$

$$\varphi_{\mathrm{PER}}(x) = [w_{N_{3_1}}\varphi_{\mathrm{PER}_{N_{3_1}}}(x), ..., w_{N_{3'_m}}\varphi_{\mathrm{PER}_{N_{3'_m}}}(x)], \tag{26}$$

where $w_{N_{j_i}} = \min_{\forall i}(E_{j_n})/E_{j_i}$ $(j = 1, 2, 3)$ with $E_{j_i}$ the *priori* knowledge of the EER performance of the development data of the subsystem. The logistic regression optimized weighting (LROW) method is used to optimize the reconstructed SVM supervector weighting coefficients. Since not all the SVM supervector reconstruction subsystems are effective when fused, we also extend the work by formulating quantitative measures to select the subsystems for fusion. The output score of the SVM classifier is computed as follows:

$$f^*(\varphi^*(x)) = \sum\nolimits_{l^*} \alpha_{l^*} K^*(\varphi^*(x), \varphi^*(x_{l^*})) + d^*, \tag{27}$$

where the reconstruction methods are represented by means of '*', and $\varphi^*(x_{l*})$ are support vectors obtained from the reconstructed SVM supervectors using the Mercer condition.

**Figure 4 Architecture of the HEPLR system.**

As mentioned previously, in the HEPLR language recognition system, the training stage is carried out between the positive set and negative set with one-versus-rest strategy.

The linear discriminant analysis-maximum mutual information (LDA-MMI) method is used to maximize the posterior probabilities of all the belief score vectors [35], using objective function [36]:

$$F_{\mathrm{MMI}}(\lambda) = \sum_{\forall i} \log \frac{p(\boldsymbol{x}_i|\lambda_{g(i)})P(g(i))}{\sum_{\forall j} p(\boldsymbol{x}_i|\lambda_j)P(j)}, \tag{28}$$

where $g(i)$ indicates the class label of $x_i$ and $P(j)$ denotes the prior probability of class $j$. Vector fusion is implemented directly as

$$\boldsymbol{x} = [w'_1 f(\varphi_{\mathrm{REL}}(x)), w'_2 f(\varphi_{\mathrm{FUN}}(x)), w'_3 f(\varphi_{\mathrm{PER}}(x))], \tag{29}$$

The probability density function $p(\boldsymbol{x}|\lambda)$ is a Gaussian Mixture Model defined on the $N$-dimensional vector $\boldsymbol{x}$:

$$p(\boldsymbol{x}|\lambda) = \sum\nolimits_{\forall m'} \omega'_{m'} \mathcal{N}(x; \mu_{m'}, \Sigma_{m'}), \tag{30}$$

The proposed homogeneous ensemble language recognition system has three advantages. First, SVM supervector reconstruction provides vector space modeling diversification for richer language

identification information. Second, in the HEPLR system, the subsystems share the same preprocessing steps for feature extraction, decoding, and expected counting, which minimizes additional computational cost. Third, fusing the reconstructed SVM supervector with the original supervector at the vector level means that more information can be retained than that given by score fusion.

## 5 Experimental setup

### 5.1 Baseline language recognition system setup

The TRAPs/NN phonotactic language recognizers developed by the BUT [37] based on phone lattices, N-gram counts, and SVM scoring are used as baseline systems. An energy-based voice activity detector that splits and removes long-duration non-speech segments from the signals is applied initially. Following this, the BUT decoders for Czech (CZ), Hungarian (HU), and Russian (RU) are applied to compute phone *posteriori* probabilities, as used in NIST LRE tasks by many groups [38,39]. The phone inventory is 43 for Czech, 59 for Hungarian, and 50 for Russian. *Posteriori* probabilities are put into the HVite decoder produced by HTK to produce phone lattices, which encode multiple hypotheses with acoustic likelihoods. The N-gram counts are produced by lattice-tool from SRILM (SRI International, Menlo Park, CA, USA) [40]. The LIBLINEAR tool [41] for multiclass SVMs with linear kernels is applied to give SVM scores. Finally, the LDA-MMI algorithm [42] is used for score calibration and fusion.

### 5.2 Training, development, and test datasets

Evaluation is carried out on the NIST LRE 2009 tasks. This data includes 41793 utterances including 30-, 10-, and 3-s nominal duration, closed condition. The NIST LRE 2009 core task recognition of is to recognize 23 languages, including Amharic, Bosnian, Cantonese, Creole, Croatian, Dari, American English, Indian English, Farsi, French, Georgian, Hausa, Hindi, Korean, Mandarin, Pashto, Portuguese, Russian, Spanish, Turkish, Ukrainian, Urdu, and Vietnamese. The evaluation involves radio broadcasts and conversational telephone speech channel conditions.

The training data comes from different sources including CallHome, CallFriend, OGI, OHSU, VOA, and the development corpora for the 2003, 2005, and 2007 NIST LRE evaluations.

About 25,000 utterances are selected randomly from VOA and 2003, 2005, and 2007 NIST LRE datasets used as development data.

### 5.3 Evaluation measures

In this work, the performance of language recognition systems is compared using: (1) EER and (2) average cost performance Cavg defined by NIST [43], which are obtained by one-versus-rest tragedy.

## 6 Experimental results and discussion

We demonstrate the effectiveness of our approaches on NIST LRE 2009 tasks under 30-, 10-, and 3-s conditions. Results are shown in Tables 1, 2, 3, 4, 5, and 6 and Figures 1, 2, 3, 4, 5, 6, 7, and 8 in the following sections. The EER and Cavg performance of individual subsystems and fusions is also shown in the tables below for reference.

**Table 1 Performance of baseline language recognition system**

| | SVM supervector dimension | 30 s | | 10 s | | 3 s | |
|---|---|---|---|---|---|---|---|
| | | EER | $C$avg | EER | $C$avg | EER | $C$avg |
| HU | 195112 | 2.44 | 2.39 | 7.54 | 7.38 | 24.23 | 23.98 |
| RU | 117649 | 2.26 | 2.06 | 6.23 | 6.07 | 20.53 | 20.38 |
| CZ | 74088 | 3.39 | 3.31 | 10.13 | 10.04 | 28.73 | 28.35 |

NIST LRE 2009 (EER and Cavg in percent).

**Table 2 Performance of relative SVM supervector reconstruction subsystem, TFLLR and RBF kernel**

| | 30 s | | 10 s | | 3 s | |
|---|---|---|---|---|---|---|
| | EER | $C$avg | EER | $C$avg | EER | $C$avg |
| HU | 2.38 | 2.40 | 7.11 | 7.17 | 20.32 | 20.49 |
| RU | 2.01 | 1.92 | 5.83 | 5.77 | 17.26 | 17.34 |
| CZ | 3.14 | 3.09 | 8.47 | 8.53 | 23.12 | 23.17 |

NIST LRE 2009 (EER and Cavg in percent).

**Table 3 Performance of functional SVM supervector reconstruction subsystems**

| | 30 s | | 10 s | | 3 s | |
|---|---|---|---|---|---|---|
| | EER | $C$avg | EER | $C$avg | EER | $C$avg |
| HU | 2.13 | 2.09 | 6.26 | 6.24 | 19.16 | 19.21 |
| RU | 2.06 | 1.98 | 5.40 | 5.34 | 17.64 | 17.71 |
| CZ | 2.86 | 2.84 | 8.36 | 8.47 | 22.89 | 23.12 |

NIST LRE 2009 (EER and Cavg in percent).

**Table 4 Performance of perturbational SVM supervector reconstruction subsystems**

| | 30 s | | 10 s | | 3 s | |
|---|---|---|---|---|---|---|
| | EER | $C$avg | EER | $C$avg | EER | $C$avg |
| HU | 2.14 | 2.10 | 6.78 | 6.84 | 20.39 | 20.49 |
| RU | 2.11 | 1.96 | 5.91 | 5.85 | 18.67 | 18.55 |
| CZ | 2.95 | 2.87 | 8.76 | 8.71 | 25.35 | 25.24 |

NIST LRE 2009 (EER and Cavg in percent).

**Table 5 Comparison of baseline SLR system and HEPLR system**

| | 30 s | | 10 s | | 3 s | |
|---|---|---|---|---|---|---|
| | EER | $C$avg | EER | $C$avg | EER | $C$avg |
| HU | 2.44 | 2.39 | 7.54 | 7.38 | 24.23 | 23.98 |
| RU | 2.26 | 2.06 | 6.23 | 6.07 | 20.53 | 20.38 |
| CZ | 3.39 | 3.31 | 10.13 | 10.04 | 28.73 | 28.35 |
| Fusion | 1.52 | 1.58 | 4.04 | 4.05 | 16.53 | 16.10 |
| HU-SSR | 1.92 | 1.84 | 5.89 | 5.91 | 18.64 | 18.73 |
| RU-SSR | 1.82 | 1.77 | 5.21 | 5.14 | 16.76 | 16.51 |
| CZ-SSR | 2.86 | 2.84 | 8.36 | 8.47 | 22.89 | 22.93 |
| Fusion-SSR | 1.39 | 1.29 | 3.63 | 3.64 | 14.79 | 14.64 |

NIST LRE 2009 (EER and Cavg in percent).

**Table 6 Comparison of real-time factor for language recognition systems**

| | Baseline | Relative SSR | Functional SSR | Perturbational SSR |
|---|---|---|---|---|
| Decoding | 0.11 | 0.11 | 0.11 | 0.11 |
| SV prod. | $2.63 \times 10^{-6}$ | 0.06 | $2.67 \times 10^{-6}$ | $2.64 \times 10^{-6}$ |
| Total | 0.11 | 0.17 | 0.11 | 0.11 |

HU frontend, NIST LRE 2009, 30-s test. CPU: Xeon E5520@2.27 GHz, RAM: 8 GB, single thread. SV prod., super vector product.

**Figure 5 Performance of relative SVM supervector reconstruction subsystem versus dimension.** NIST LRE 2009, 30-s, HU frontend (EER and Cavg in percent). **(a)** TFLLR kernel. **(b)** RBF kernel. **(c)** TFLLR and RBF kernel.

**Figure 6 Performance of functional SVM supervector reconstruction subsystems.** NIST LRE 2009, 30-s, HU frontend. '+' indicated fusion (EER and Cavg in percent).

**Figure 7 Performance of perturbational SVM supervector reconstruction subsystems.** NIST LRE 2009, 30-s, HU frontend (EER and Cavg in percent). **(a)** Approach 1. **(b)** Approach 2. **(c)** Approach 3. **(d)** Approach 4.

**Figure 8 DET curves of baseline system and HEPLR system for NIST LRE 2009.**

### 6.1 Baseline PRVSM system

Table 1 shows EER and Cavg performance for the NIST LRE 2009 language recognition tasks using the baseline subsystems. In this work, the dimension of the possible 3-gram SVM supervector produced by the single Hungarian (HU) phone recognizer with 58 phones is $58 \times 58 \times 58 = 195112$. SVM supervector dimensions for the Russian (RU) and Czech (CZ) recognizers are 117649 and 74088, respectively.

### 6.2 Relative SVM supervector reconstruction

In this paper, 13,000 conversations which are randomly selected from the 40 languages of the 2003, 2005, and 2007 NIST LRE and VOA, CallHome, and CallFriend Corpora. These are used as the dataset to build the relative SVM supervector reconstructor.

Figure 5 shows the performance of the relative SVM supervector reconstruction subsystems whose SVM classifier uses the TFLLR and RBF kernel, respectively. Figure 5 also shows the performance of the relative SVM supervector reconstruction subsystems whose SVM classifier uses fusion of TFLLR and RBF kernel. Table 2 shows that the performance of the relative SVM supervector reconstruction subsystem is better than baseline and increases slowly with increasing size of the datum dataset. A lower dimension relative SVM supervector can be chosen to trade off performance and computation cost. The language recognition performance for short utterances is significantly improved in the relative SSR subsystem compared to the baseline. The original supervector can not describe short utterances precisely because of insufficient phoneme data, while the relative reconstructed SVM supervectors use the relationship between a short utterance and a large set of datum utterances, which is a richer representation. The experimental results show that the relative SSR subsystem outperforms the baseline and can obtain better performance with relative feature using a low-dimension SVM supervector.

### 6.3 Functional SVM supervector reconstruction

The language recognition results of functional SVM supervector reconstruction subsystems are given in Figure 6 and Table 3. The results using this approach were similar to or slightly worse than the baseline system in 30-s test condition, but outperform the baseline system in the 10- and 3-s test conditions.

### 6.4 Perturbational SVM supervector reconstruction

Figure 7 and Table 4 describe the results of the four perturbational methods. Overall, approach 2 yielded better results (2.20%, 6.59%, 20.93% EER) than the other approaches. The perturbative SVM supervector reconstruction subsystems performed consistently better than the baseline subsystem; particularly, those based on approach 2 performed better than the others. We hypothesize that approach 2 outperforms other perturbation methods because the distribution of the perturbation better matches the distribution of the noise. The perturbation approach adds robustness to the language modeling.

### 6.5 SVM supervector reconstruction

From the experimental results and discussion, it can be concluded that some of the reconstruction methods (relative SSR) are better at identifying the language of the short utterance and others (functional SSR and perturbational SSR) are better at recognizing long utterances. Because these errors are not highly correlated, we can fuse these results together to harness the complementary behavior among subsystems and improve the language recognition performance.

Figure 8 shows DET curves of the baseline system versus the HEPLR system for NIST LRE 2009. Table 5 gives the corresponding performance numbers for all configurations. These results show that the SSR approaches proposed in this paper outperformed the baseline system in terms of EER and Cavg when considering complete fusions for the subsystems.

### 6.6 Real-time factors

Table 6 shows the real-time (RT) factors of each part of SSR system. From Table 6, we can see that decoding is the dominant part. Compared to PR-VSM baseline system, the computational cost increases about 1.5 times for the relative SSR and barely no increases for the functional SSR and perturbational SSR.

## 7 Computational cost

Let $F$, $M$, and $M_{\text{datum}}$ denote the dimension of the phonotactic feature supervector of an utterance, the number of utterances of training dataset, and the datum dataset, respectively. And let $c_\varphi$ denote the computation cost of the mapping from $x$ to $\varphi(x)$, and $c_{\text{modeling}}(F, M)$ denotes the computational cost of modeling the languages, which relate to $F$ and $M$. Then, the computational cost of the baseline system is

$$c_{\text{baseline}} = M \cdot c_\varphi + c_{\text{modeling}}(F, M) \tag{31}$$

$$c_\varphi = c_{\text{Pre-Processing}} + c_{\text{FeatureExtract}} + c_{\text{Decoding}} + c_{\text{N-gramCounting}}, \tag{32}$$

where $c_{\text{Pre-Processing}}$, $c_{\text{FeatureExtract}}$, $c_{\text{Decoding}}$, and $c_{\text{N-gramCounting}}$ denote the computational cost of preprocessing, feature extracting, decoding, and $N$-gram counting, respectively.

### 7.1 Relative SVM supervector reconstruction

Let $c_{\text{inner}}(F)$ denote the computation cost of the inner product of the two $F$ dimensional supervectors. Then, the computational cost of the relative SVM supervector reconstruction system is computed as

$$c_{\text{REL}} = M \cdot c_{\varphi} + M_{\text{datum}} \cdot c_{\varphi} + c_{\text{modeling}}(M_{\text{datum}}, M) + M \cdot M_{\text{datum}} \cdot c_{\text{inner}}(F) \tag{33}$$

Usually, $c_{\text{modeling}}(M_{\text{datum}}, M) < c_{\text{modeling}}(F, M) \ll M \cdot c_{\varphi}$, $M \cdot M_{\text{datum}} \cdot c_{\text{inner}}(F) \ll M \cdot c_{\varphi}$, so

$$\frac{c_{\text{REL}}}{c_{\text{baseline}}} \approx \frac{M \cdot c_{\varphi} + M_{\text{datum}} \cdot c_{\varphi}}{M \cdot c_{\varphi}} = 1 + \frac{M_{\text{datum}}}{M} \tag{34}$$

In this paper, $M = 30996$, when considering RU frontend, then $F = 117649$. When $M_{\text{datum}} = 13000$, $c_{\text{REL}}/c_{\text{baseline}} = 41.94\%$. That means that the relative SVM supervector reconstruction system takes 41.94% extra computation and achieves a 11.84%, 6.42%, and 15.92% relative improvements, respectively, for 30-, 10-, and 3-s compared to the baseline.

### 7.2 Functional SVM supervector reconstruction

Let $c_{\varphi_{\text{FUN}}}$ denote the computational cost of mapping $\varphi(x)$ to $\varphi_{\text{FUN}}(x)$. Then, the computational cost of the functional SVM supervector reconstruction system is computed as

$$c_{\text{FUN}} = M \cdot c_{\varphi} + M \cdot c_{\varphi_{\text{FUN}}} + c_{\text{modeling}}(F, M), \tag{35}$$

because preprocessing, feature extracting, decoding, and $N$-gram counting are more complex than the functional computation in this paper, so $M \cdot c_{\varphi_{\text{FUN}}} \ll M \cdot c_{\varphi}$. The computational cost of modeling the languages can be considered equal to the baseline. For RU frontend, the functional SVM supervector reconstruction system takes almost no extra computation and achieves 8.84%, 13.32%, and 14.76% relative improvements, respectively, for 30-, 10-, and 3-s compared to the baseline.

### 7.3 Perturbational SVM supervector reconstruction

Let $c_{\varphi_{\text{PER}}}$ denote the computational cost of adding perturbation to $\varphi(x)$. Then, the computational cost of the functional SVM supervector reconstruction system is computed as

$$c_{\text{PER}} = M \cdot c_{\varphi} + M \cdot c_{\varphi_{\text{PER}}} + c_{\text{modeling}}(F, M), \tag{36}$$

because preprocessing, feature extracting, decoding, and $N$-gram counting are more complex than the perturbational computation in this paper, so $M \cdot c_{\varphi_{\text{PER}}} \ll M \cdot c_{\varphi}$. The computational cost of modeling the languages can be considered equal to the baseline. For RU frontend, the perturbational SVM supervector reconstruction system takes almost no extra computation and achieves 6.63%, 5.13%, and 9.05% relative improvements, respectively, for 30-, 10-, and 3-s compared to the baseline.

## 8 Conclusions

In this article, we investigate a strategy of SVM supervector reconstruction to provide vector space modeling diversification to improve the performance and robustness of language recognition tasks with very low additional computational cost. A variety of SVM supervector reconstruction methods are employed to develop the diversified SVM supervectors. Reconstruction methods include relative SSR, perturbational SSR, and functional SSR. Relative SSR method uses the relationship of an utterance and a datum set to present the utterance.

Perturbational SSR reconstructs the SVM supervector to a slightly perturbational version and improves the language recognition performance. Functional SSR can derive effective kernel mixtures and get robust language model. The approaches do not involve significant additional computation compared to a baseline phonotactic system, but represents a way to extract more information from existing decodings.

Experimental results of the proposed HEPLR system on the NIST LRE 2009 evaluation set show better performance than the baseline system. When we fuse the three subsystems at the score level for further improvements, we achieve 1.39%, 3.63%, and 14.79% EER for the 30-, 10-, and 3-s closed-set test conditions, respectively. This corresponds to 6.06%, 10.15%, and 10.53% relative improvements.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

This project is supported by the National Natural Science Foundation of China under grant nos. 61370034, 61273268, and 61403224.

## References

1. MP Lewis, Ethnologue: languages of the world, 16th edn. (SIL International, 2009). http://www.ethnologue.com. Accessed 17 Jan 2007
2. VW Zue, JR Glass, Conversational interfaces: advances and challenges. Proc. IEEE **88**(8), 1166?1180 (2000)
3. MA Zissman, Comparison of four approaches to automatic language identification of telephone speech. IEEE Trans. Speech Audio Proc. **4**(1), 31?44 (1996)
4. YK Muthusamy, E Barnard, RA Cole, Reviewing automatic language identification. IEEE Signal Process. Mag. **11**(4), 33?41 (1994)
5. PA Torres-Carrasquillo, E Singer, MA Kohler, RJ Greene, DA Reynolds, JR Deller, Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Denver, 16?20 Sept 2002, pp. 33?36
6. P Matejka, O Glembek, F Castaldo, MJ Alam, O Plchot, P Kenny, L Burget, J Cernocky, Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification. in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, 22?27 May 2011, pp. 4828?4831
7. N Dehak, PJ Kenny, R Dehak, P Dumouchel, P Ouellet, Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Language Process. **19**(4), 788?798 (2011)
8. LF Dharo Enriquez, O Glembek, O Plchot, P Matejka, M Soufifar, Rd Cordoba Herralde, J Cernocky, Phonotactic language recognition using i-vectors and phoneme posteriogram counts. in *Proceedings of INTERSPEECH*, Oregon, 9?13 Sept 2012, pp. 42?45
9. D Garcia-Romero, C Espy-Wilson, Joint factor analysis for speaker recognition reinterpreted as signal coding using overcomplete dictionaries. in *Proceedings of Odyssey?The Speaker and Language Recognition Workshop*, Brno, 7?11 July 2010, pp. 117?124

10. G Heigold, H Ney, R Schluter, S Wiesler, Discriminative training for automatic speech recognition: modeling, criteria, optimization, implementation, and performance. IEEE Signal Process. Mag. **29**(6), 58?69 (2012)

11. WM Campbell, JP Campbell, DA Reynolds, E Singer, PA Torres-Carrasquillo, Support vector machines for speaker and language recognition. Comput. Speech Language **20**(2), 210?229 (2006)

12. H Li, B Ma, K-A Lee, Spoken language recognition: from fundamentals to practice. Proc. IEEE **101**(5), 1136?1159 (2013)

13. E Singer, P Torres-Carrasquillo, D Reynolds, A McCree, F Richardson, N Dehak, D Sturim, The MITLL NIST LRE 2011 language recognition system. in *Proceedings of Odyssey?The Speaker and Language Recognition Workshop*, Singapore, 25?28 June 2012, pp. 209?215

14. LJ Rodriguez-Fuentes, M Penagarikano, A Varona, M Diez, G Bordel, A Abad, D Martinez, J Villalba, A Ortega, E Lleida, The BLZ submission to the NIST 2011 LRE: data collection, system development and performance. in *Proceedings of INTERSPEECH*, Oregon, 9?13 Sept 2012, pp. 38?41

15. M Penagarikano, A Varona, LJ Rodriguez-Fuentes, M Diez, G Bordel, University of the Basque Country (EHU) systems for the 2011 NIST language recognition evaluation. in *Proceedings of the NIST 2011 LRE Workshop*, Gaithersburg, 6?7 Dec 2011, pp. 1?5

16. F Zheng, G Zhang, Z Song, Comparison of different implementations of MFCC. J Comput. Sci. Technol. **16**(6), 582?589 (2001)

17. A Zolnay, R Schl?ter, H Ney, Acoustic feature combination for robust speech recognition. in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, 18?23 March 2005, pp. 457?460

18. V Hubeika, L Burget, P Matejka, P Schwarz, Discriminative training and channel compensation for acoustic language recognition. in *Proceedings of INTERSPEECH*, Brisbane, 22?26 Sept 2008, pp. 301?304

19. H Li, B Ma, C-H Lee, A vector space modeeling approach to spoken language identification. IEEE Trans. Audio Speech Language Process. **15**(1), 271?284 (2007)

20. KC Sim, H Li, On acoustic diversification front-end for spoken language identification. IEEE Trans. Audio Speech Language Process. **16**(5), 1029?1037 (2008)

21. N Morgan, H Bourlard, Continuous speech recognition using multilayer perceptrons with hidden Markov models. in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kobe, 18?22 Nov 1990, pp. 413?416

22. PA Torres-Carrasquillo, Language identification using Gaussian mixture models. PhD thesis, Michigan State University, 2002

23. W-W Liu, M Cai, H Yuan, J Xu, J Liu, W-Q Zhang, Phonotactic Language Recognition Based on DNN-HMM Acoustic Model. in *Proceedings of the IEEE International Symposium on Chinese Spoken Language Processing (ISCSLP)*, Singapore, 12?14 Sept 2014, pp. 148?152

24. W-W Liu, W-Q Zhang, Y Shi, A Ji, J Xu, J Liu, Improved phonotactic language recognition based on RNN feature reconstruction. in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, 4?9 May 2014, pp. 5359?5363

25. W-W Liu, W-Q Zhang, Z Li, J Liu, Parallel absolute-relative feature based phonotactic language recognition. in *Proceedings of INTERSPEECH*, Lyon, 25?29 Aug 2013, pp. 2474?2478

26. WM Campbell, JP Campbell, DA Reynolds, E Singer, PA Torres-Carrasquillo, Support vector machines for speaker and language recognition. Comput. Speech Language **20**(2?3), 210?229 (2006)

27. JL Gauvain, A Messaoudi, H Schwenk, Language recognition using phone lattices. in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Jesu Land, 4?8 Oct 2004, pp. 1283?1286

28. WM Campbell, JP Campbell, DA Reynolds, DA Jones, TR Leek, Phonetic speaker recognition with support vector machines. Adv. Neural Inf. Process. Syst. **16**, 1377?1384 (2003)

29. B Scholkopf, J Weston, E Eskin, C Leslie, WS Noble, Kernel approach for learning from almost orthogonal patterns. in *Proceedings of the European Conference on Machine Learning (ECML)*, Helsinki, 19?23 Aug 2002, pp. 511?528

30. M Wang, S Chen, Enhanced FMAM based on empirical kernel map. IEEE Trans. Neural Networks **16**(3), 557?564 (2005)

31. H Xiong, MNS Swamy, MO Ahmad, Optimizing the kernel in the empirical feature. IEEE Trans. Neural Networks **16**(2), 460?474 (2005)

32. P Matejka, P Schwarz, J Cernockỳ, P Chytil, Phonotactic language identification using high quality phoneme recognition. in *Proceedings of INTERSPEECH*, Lisbon, 4?8 Sept 2005, pp. 2237?2240

33. P Vincent, H Larochelle, I Lajoie, Y Bengio, P-A Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Machine Learning Res. **11**, 3371?3408 (2010)

34. H Li, B Ma, C-H Lee, A vector space modeling approach to spoken language identification. IEEE Trans. Audio Speech Language Process. **15**(1), 271?284 (2007)

35. P Matejka, L Burget, O Glembek, P Schwarz, V Hubeika, M Fapso, T Mikolov, O Plchot, BUT system description for NIST LRE 2007. in *Proceedings of the NIST Language Recognition Evaluation Workshop*, Orlando, 11?12 Dec 2007, pp. 1?5

36. D Povey, PC Woodland, Improved discriminative training techniques for large vocabulary continuous speech recognition. in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, 22?27 May 2011, pp. 45?48

37. P Schwarz, Phoneme recognition based on long temporal context. PhD thesis, Brno University of Technology, 2009

38. Z Jancik, O Plchot, N Brummer, L Burget, O Glembek, V Hubeika, M Karafiat, P Matejka, T Mikolov, A Strasheim, J Cernocky, Data selection and calibration issues in automatic language recognition-investigation with but-agnitio NIST LRE 2009 system. in *Proceedings of Odyssey? The Speaker and Language Recognition Workshop*, Brno, 7?11 July 2010, pp. 215?221

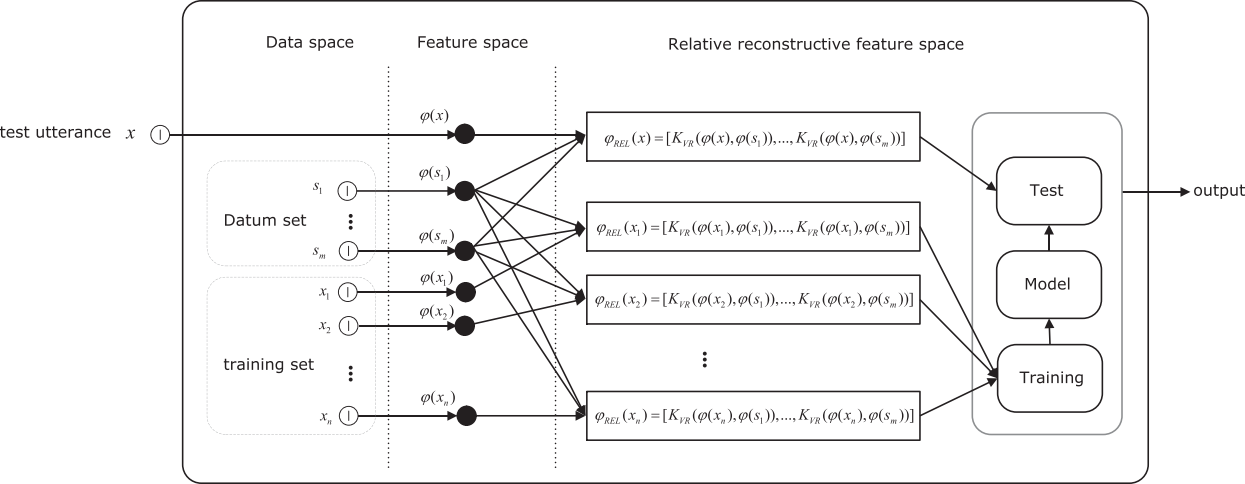
39. PA Torres-Carrasquillo, E Singer, T Gleason, A McCree, DA Reynolds, F Richardson, D Sturim, The MITLL NIST LRE 2009 language recognition system. in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dalas, 14?19 March 2010, pp. 4994?4997

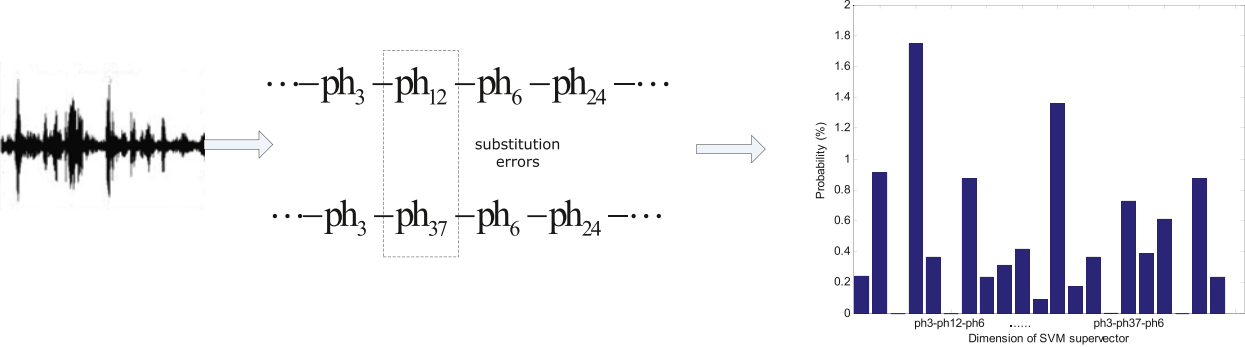
40. A Stolcke, SRILM - an extensible language modeling toolkit. (SRI International, 2002) http://www.speech.sri.com/projects/srilm/. Accessed 3 April 2002

41. R Collobert, S Bengio, SVMTorch: support vector machines for large-scale regression problems. J. Machine Learning Res. **1**, 143?160 (2001)

42. W-Q Zhang, T Hou, J Liu, Discriminative score fusion for language identification. Chin. J. Electron. **19**, 124?128 (2010)

43. The 2009 NIST language recognition evaluation plan. (U.S. Department of Commerce) http://www.itl.nist.gov/iad/mig/tests/lang/2009/. Accessed 1 April 2009

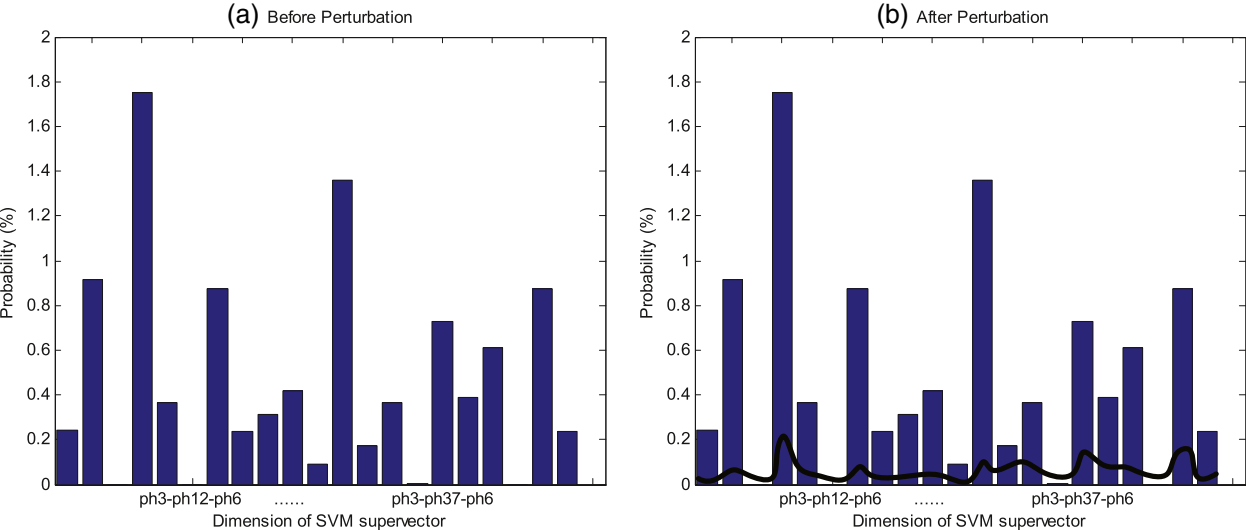

Data space
Feature space
Relative reconstructive feature space
test utterance $x$
$\varphi(x)$
$s_1$
$\varphi(s_1)$
Datum set
$s_m$
$\varphi(s_m)$
$x_1$
$\varphi(x_1)$
$x_2$
$\varphi(x_2)$
training set
$x_n$
$\varphi(x_n)$
$\varphi_{REL}(x) = [K_{VR}(\varphi(x), \varphi(s_1)), ..., K_{VR}(\varphi(x), \varphi(s_m))]$
$\varphi_{REL}(x_1) = [K_{VR}(\varphi(x_1), \varphi(s_1)), ..., K_{VR}(\varphi(x_1), \varphi(s_m))]$
$\varphi_{REL}(x_2) = [K_{VR}(\varphi(x_2), \varphi(s_1)), ..., K_{VR}(\varphi(x_2), \varphi(s_m))]$
$\varphi_{REL}(x_n) = [K_{VR}(\varphi(x_n), \varphi(s_1)), ..., K_{VR}(\varphi(x_n), \varphi(s_m))]$
Test
Model
Training
output

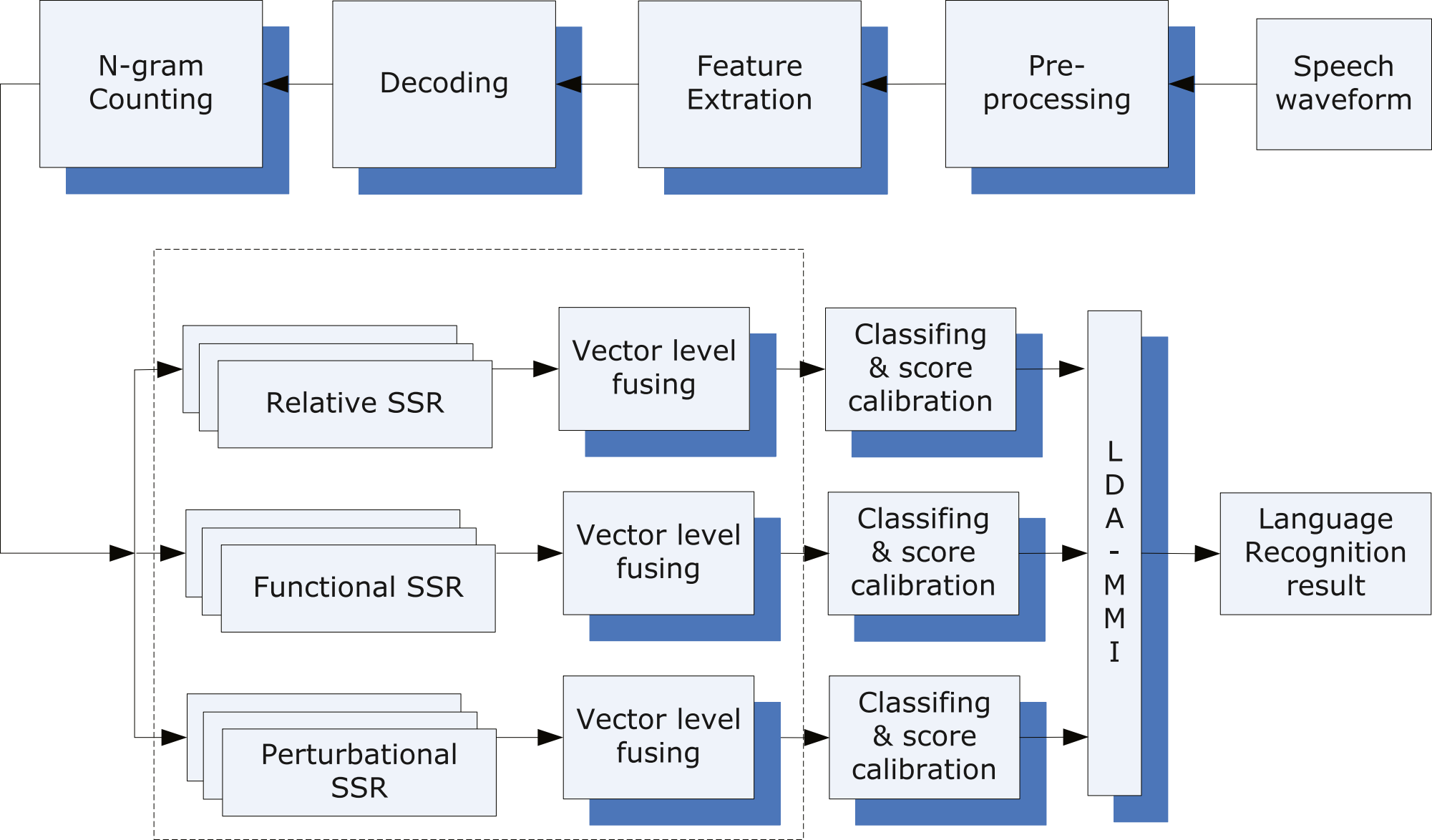


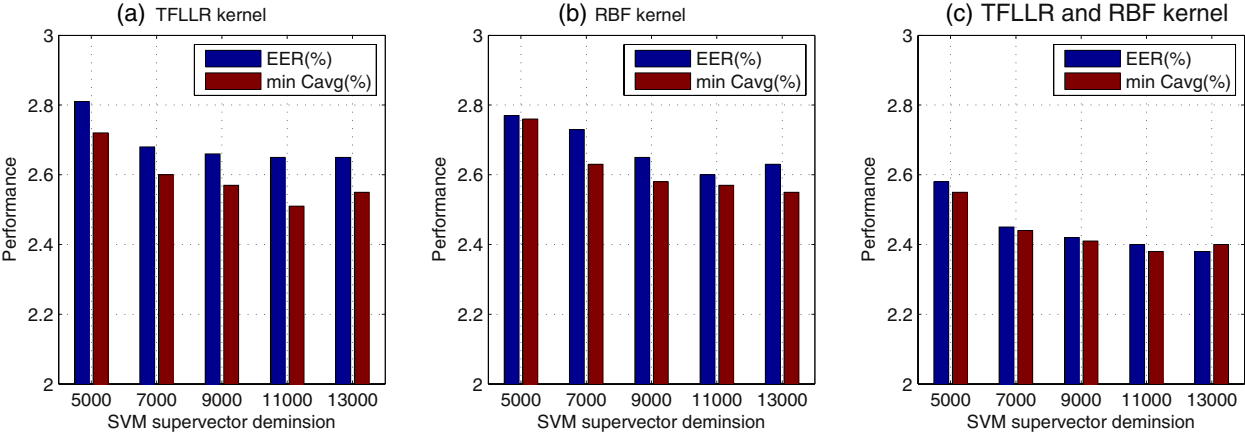

Speech wave

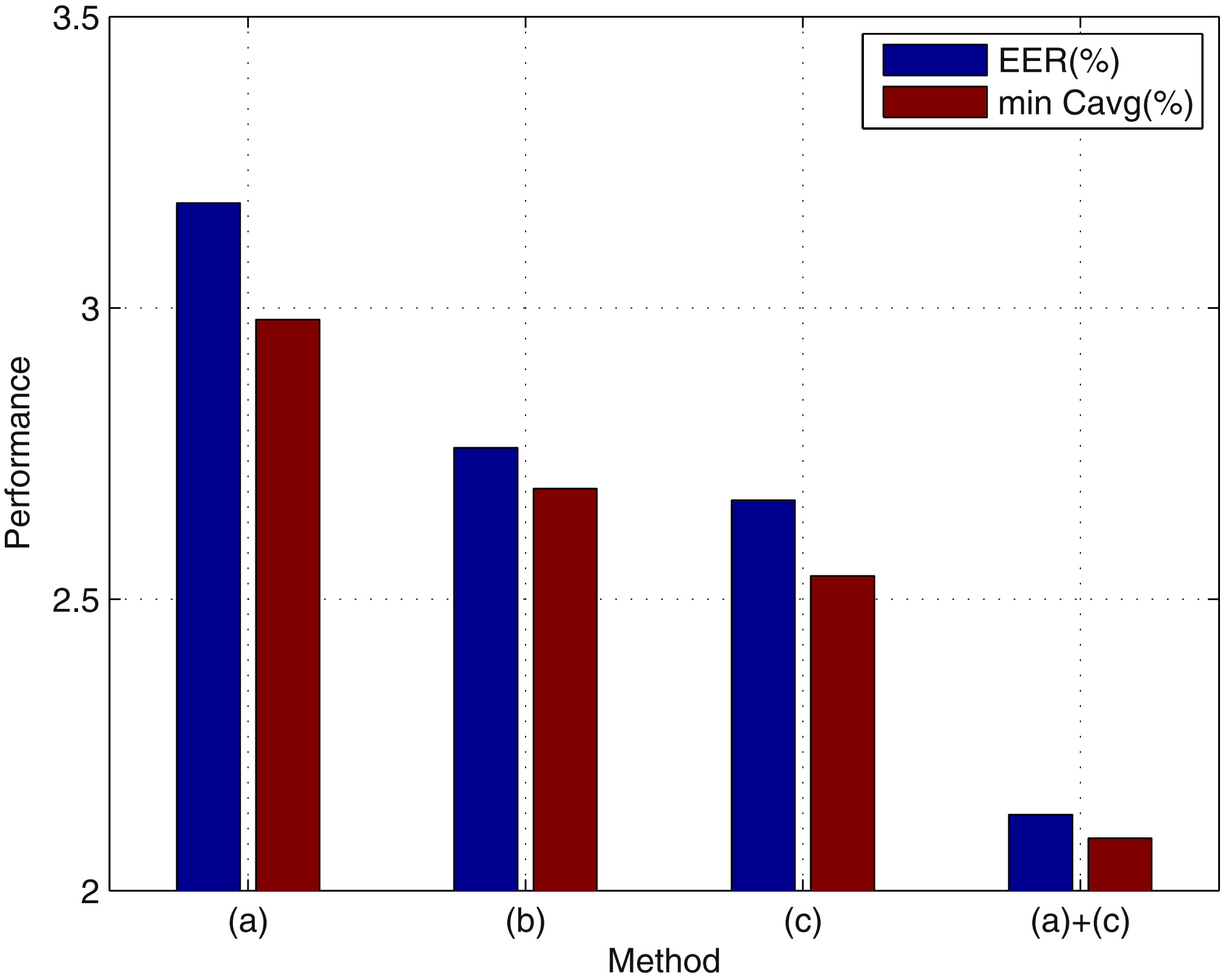
Phoneme string
(Lattice)

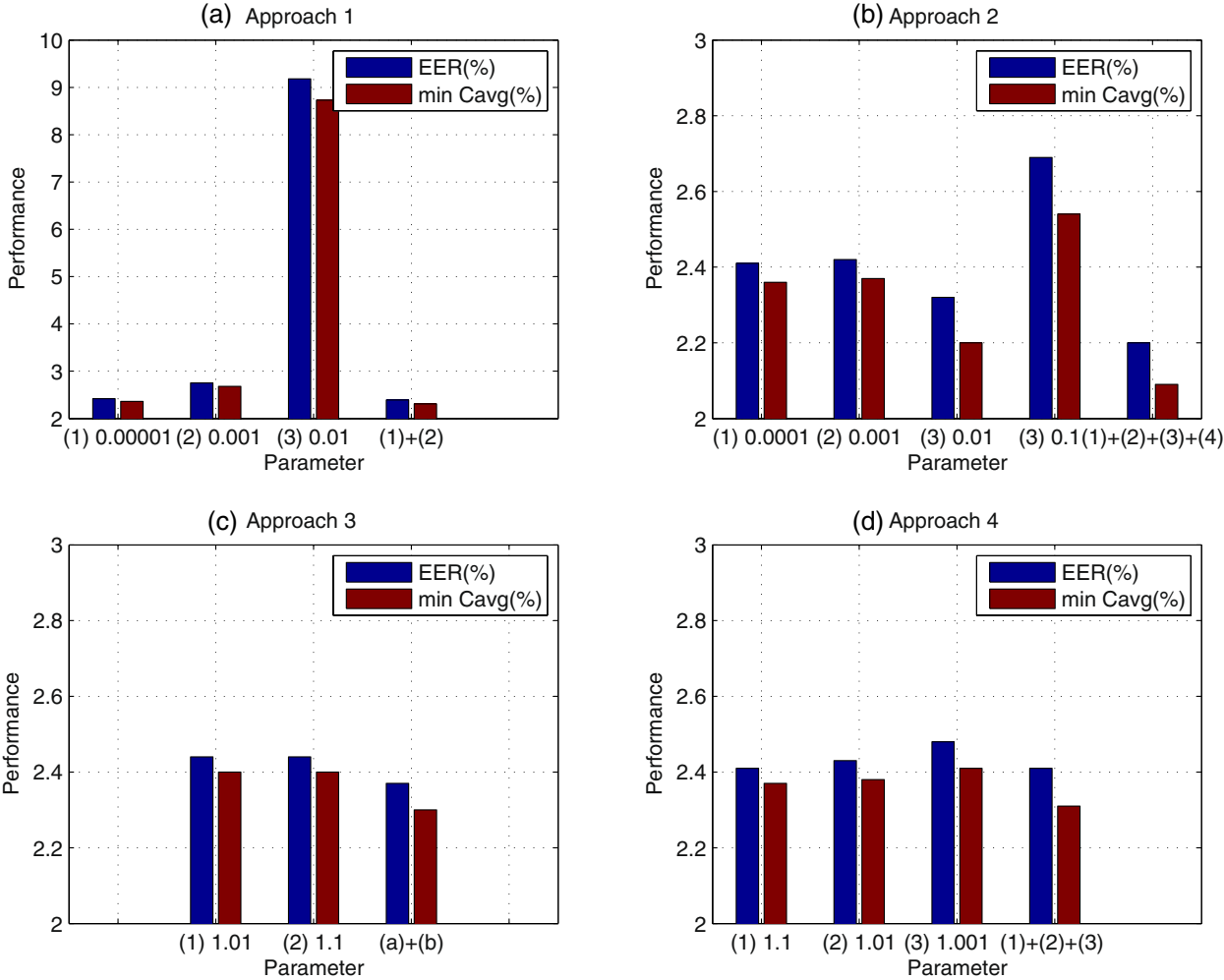
SVM supervector

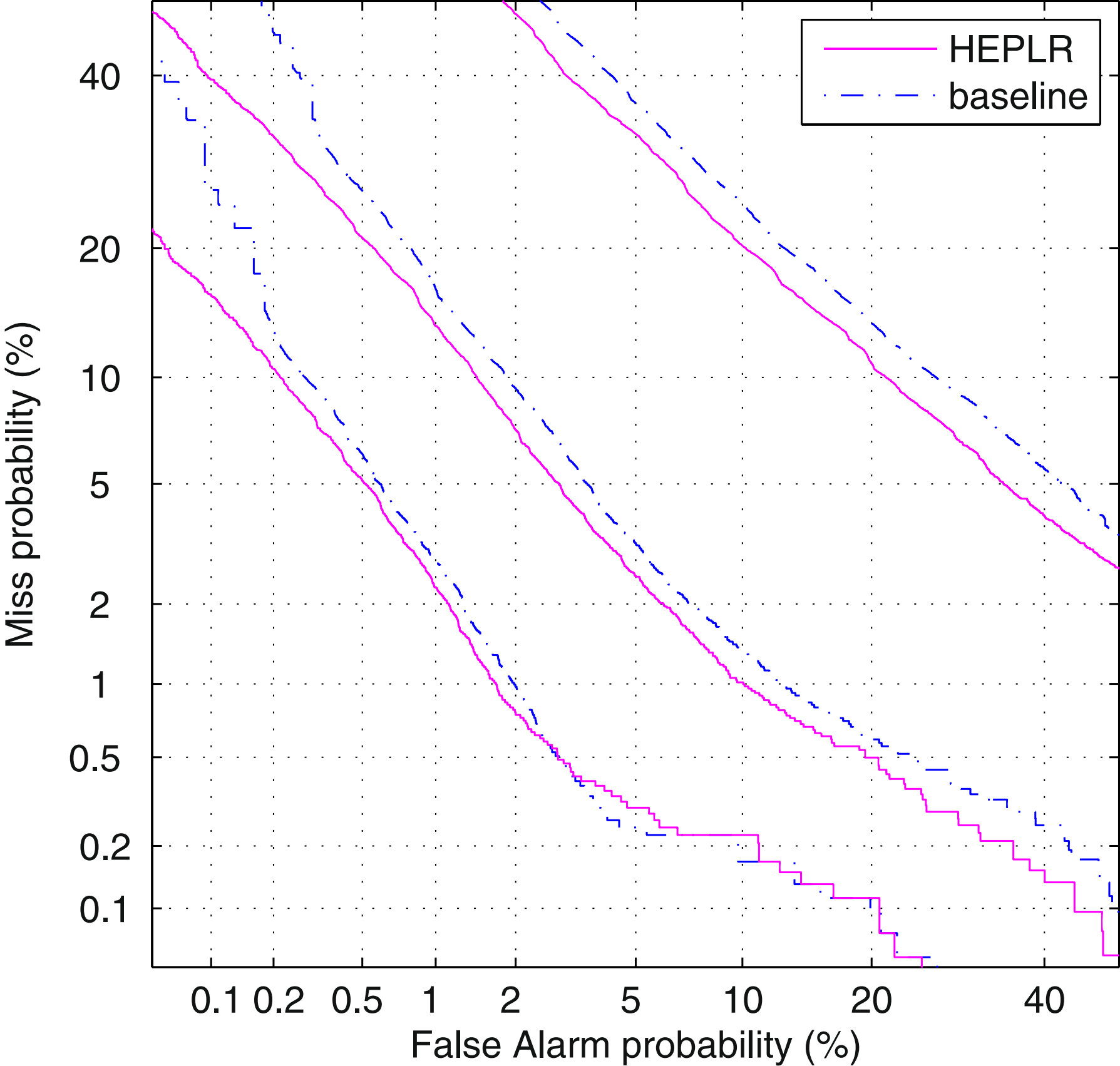

(a) Before Perturbation
(b) After Perturbation
Probability (%)
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
ph3-ph12-ph6
.....
ph3-ph37-ph6
Dimension of SVM supervector


N-gram Counting
Decoding
Feature Extration
Pre-processing
Speech waveform
Relative SSR
Functional SSR
Perturbational SSR
Vector level fusing
Vector level fusing
Vector level fusing
Classifing & score calibration
Classifing & score calibration
Classifing & score calibration
L D A - M M I
Language Recognition result


(a) TFLLR kernel
(b) RBF kernel
(c) TFLLR and RBF kernel
EER(%)
min Cavg(%)
Performance
SVM supervector deminsion
3
2.8
2.6
2.4
2.2
2
5000
7000
9000
11000
13000


EER(%)
min Cavg(%)
3.5
3
2.5
2
Performance
(a)
(b)
(c)
(a)+(c)
Method


(a) Approach 1
EER(%)
min Cavg(%)
Performance
10
9
8
7
6
5
4
3
2
(1) 0.00001
(2) 0.001
(3) 0.01
(1)+(2)
Parameter
(b) Approach 2
EER(%)
min Cavg(%)
Performance
3
2.8
2.6
2.4
2.2
2
(1) 0.0001
(2) 0.001
(3) 0.01
(3) 0.1
(1)+(2)+(3)+(4)
Parameter
(c) Approach 3
EER(%)
min Cavg(%)
Performance
3
2.8
2.6
2.4
2.2
2
(1) 1.01
(2) 1.1
(a)+(b)
Parameter
(d) Approach 4
EER(%)
min Cavg(%)
Performance
3
2.8
2.6
2.4
2.2
2
(1) 1.1
(2) 1.01
(3) 1.001
(1)+(2)+(3)
Parameter


HEPLR
baseline
Miss probability (%)
40
20
10
5
2
1
0.5
0.2
0.1
0.1
0.2
0.5
1
2
5
10
20
40
False Alarm probability (%)