Sparse Kernel Machines with Empirical Kernel Maps for PLDA Speaker Verification

Wei RAO and Man-Wai MAK

Dept. of Electronic and Information Engineering,
The Hong Kong Polytechnic University
ellen.wei-rao@connect.polyu.hk, enmwmak@polyu.edu.hk

Abstract

Previous studies have demonstrated the benefits of PLDA-SVM scoring with empirical kernel maps for i-vector/PLDA speaker verification. The method not only performs significantly better than the conventional PLDA scoring and utilizes the multiple enrollment utterances of target speakers effectively, but also opens up opportunity for adopting sparse kernel machines in PLDA-based speaker verification systems. This paper proposes taking the advantages of empirical kernel maps by incorporating them into a more advanced kernel machine called relevance vector machines (RVMs). The paper reports extensive analyses on the behaviors of RVMs and provides insight into the properties of RVMs and their applications in i-vector/PLDA speaker verification. Results on NIST 2012 SRE demonstrate that PLDA-RVM outperforms the conventional PLDA and that it achieves a comparable performance as PLDA-SVM. Results also show that PLDA-RVM is much sparser than PLDA-SVM.

Keywords: Relevance vector machines, Empirical kernel maps,
Probabilistic linear discriminant analysis, I-vectors, NIST SRE.

1. Introduction

Nowadays, utilizing i-vectors [1] as features and probabilistic linear discriminant analysis (PLDA) [2, 3, 4] as back-end classifiers are the most popular strategies in speaker verification. Likelihood ratio (LR) scores from two hypotheses are used as verification decisions in i-vector/PLDA systems. Given a test i-vector and a target-speaker i-vector, the two hypotheses are that the test i-vector and the target-speaker i-vector are from the same speaker and that these two i-vectors are from two different speakers. Accordingly, no other i-vectors are involved in the computation of the LR score. this scoring method *implicitly* uses background information through the universal background model (UBM) [5], total variability matrix [1], and PLDA's factor loading matrix. Although this LR scoring method is computationally efficient, the implicit use of background information is a drawback.

To address the limitation of these scoring methods, PLDA-SVM equipped with empirical kernel maps (EKMs) and support vector machines (SVMs) was proposed to take the background speaker information *explicitly* into consideration during the scoring process [6, 7]. This method captures the discrimination between a target-speaker and non-target-speakers in the SVM weights. Specifically, for each target speaker, an empirical score space with dimension equal to the number of enrollment i-vectors of this target speaker is defined by using the idea of empirical kernel maps [8, 9, 10]. Given an i-vector, a score vector living in this space is formed by computing the LR scores of this i-vector with respect to each of the enrollment i-vectors. A speaker-dependent SVM – referred to as PLDA-SVM – can then be trained using the training score vectors. During verification, given a test i-vector and the target-speaker under test, the LR scores are mapped to a score vector, which is then fed to the target-speaker's SVM to obtain the final test score. The empirical kernel map presented in this paper is related to the anchor model [11, 12, 13, 14]. However, in the anchor model, a test utterance is projected onto a space represented by a set of reference speakers *unrelated* to the target-speakers, whereas in the empirical kernel map, the test utterance is projected onto an empirical feature space specific to the claimed speaker.

Compared with previous speaker recognition evaluations (SRE), NIST 2012 SRE [15] introduces some new protocols that help researchers to enhance the performance of speaker verification systems. One of the new protocols is that some target speakers have multiple enrollment utterances. PLDA-SVM with empirical kernel maps is not only a novel way of incorporating multiple enrollment i-vectors in the scoring process, but also opens up opportunity for adopting sparse kernel machines in PLDA-based speaker verification systems. Accordingly, this paper proposes to incorporate the empirical kernel maps into a sparse kernel machine known as the relevance vector machine (RVM) [16]. The main difference between SVM and RVM lies in the learning methods. The former is based on structural risk minimization, whereas the latter is based on a fully probabilistic framework. RVMs do not suffer from the limitations of SVM [16], but can obtain a comparable performance as SVM.

RVMs have been applied to speaker identification. For example, Tang et al. [17] compared the performance of GMM-UBM, SVM, and RVM for text-independent speaker identification under adverse far-field recording conditions with extremely short utterances. The input features of the RVMs in [17] are MFCC, whereas the input to the RVM in this paper is PLDA score vectors.

Comparing with our earlier work [18], we provide additional experiments and analyses in this paper. In summary, this paper has three objectives:

1. utilizing speaker-dependent score spaces as opposed to the fixed speaker-independent score space used by the conventional anchor model;
2. investigating the property of empirical kernel maps in SVMs, RVM regressions, and RVM classifications;
3. comparing PLDA-RVM with PLDA-SVM from three perspectives: evaluation performance, sparsity, and actual computation time.

The paper is organized as follows. Section 2 describes the idea of empirical kernel maps in PLDA speaker verification. Section 3 presents the PLDA-SVM scoring with empirical kernel maps. Section 4 introduces RVM regression and RVM classification using empirical kernel maps. In Sections 5 and 6, we report results based on NIST 2012 SRE. Section 7 summarizes the findings of this work.

2. Empirical Kernel Maps

Given a length-normalized [3] test i-vector $\mathbf{x}_t$ and target-speaker's i-vector $\mathbf{x}_s$, the Gaussian-PLDA likelihood ratio score can be computed as follows [2, 3, 6]:

$$\begin{aligned} S_{\text{LR}}(\mathbf{x}_t, \mathbf{x}_s) &= \frac{P(\mathbf{x}_t, \mathbf{x}_s|\text{same speaker})}{P(\mathbf{x}_t, \mathbf{x}_s|\text{different speakers})} \\ &= \text{const} + \mathbf{x}_t^{\mathsf{T}}\mathbf{Q}\mathbf{x}_t + \mathbf{x}_s^{\mathsf{T}}\mathbf{Q}\mathbf{x}_s + 2\mathbf{x}_t^{\mathsf{T}}\mathbf{P}\mathbf{x}_s, \end{aligned} \tag{1}$$

where

$$\begin{aligned} \mathbf{P} &= \mathbf{\Lambda}^{-1}\mathbf{\Gamma}(\mathbf{\Lambda} - \mathbf{\Gamma}\mathbf{\Lambda}^{-1}\mathbf{\Gamma})^{-1}; \quad \mathbf{\Lambda} = \mathbf{W}\mathbf{W}^{\mathsf{T}} + \mathbf{\Sigma} \\ \mathbf{Q} &= \mathbf{\Lambda}^{-1} - (\mathbf{\Lambda} - \mathbf{\Gamma}\mathbf{\Lambda}^{-1}\mathbf{\Gamma})^{-1}; \quad \mathbf{\Gamma} = \mathbf{W}\mathbf{W}^{\mathsf{T}}. \end{aligned} \tag{2}$$

In Eq. 2, $\mathbf{W}$ is the factor loading matrix and $\mathbf{\Sigma}$ is the covariance of the PLDA model. Eq. 1 and Eq. 2 suggest that PLDA-LR scoring uses the information of background speakers implicitly through $\mathbf{W}$ and $\mathbf{\Sigma}$. To make better use of multiple enrollment utterances of target speakers and to explicitly use the information of background speakers, we have recently proposed a speaker-dependent discriminative model that incorporates the empirical kernel maps for scoring [6, 7, 18]. We refer to the mapping from i-vectors to PLDA score vectors as empirical kernel maps (EKMs).

Assume that target-speaker s has H_s enrollment utterances and that each enrollment utterance leads to one i-vector. Then, H_s i-vectors will be obtained. In case the speaker provides one or a very small number of enrollment utterances only, we can apply an utterance partitioning technique [19] to produce multiple i-vectors from his/her enrollment utterance. Denote these i-vectors as:

$$\mathcal{X}_s = \{\mathbf{x}_{s,1}, \ldots, \mathbf{x}_{s,j}, \ldots, \mathbf{x}_{s,H_s}\}. \tag{3}$$

Let's denote the set of non-target-speaker i-vectors as:[1]

$$\mathcal{X}_b = \{\mathbf{x}_{b,1}, \ldots, \mathbf{x}_{b,i}, \ldots, \mathbf{x}_{b,B}\}. \tag{4}$$

Therefore, $\mathcal{X} = \{\mathcal{X}_s, \mathcal{X}_b\}$ is the training set for target-speaker s. Because target speakers have different numbers of enrollment utterances, the dimension of the resulting PLDA score vectors is different for different speakers.

The empirical kernel map is defined as:

$$\overrightarrow{S}_{\text{LR}}(\mathbf{x}, \mathcal{X}_s) = \begin{bmatrix} S_{\text{LR}}(\mathbf{x}, \mathbf{x}_{s,1}) \\ S_{\text{LR}}(\mathbf{x}, \mathbf{x}_{s,2}) \\ \vdots \\ S_{\text{LR}}(\mathbf{x}, \mathbf{x}_{s,H_s}) \end{bmatrix} \tag{5}$$

where $S_{\text{LR}}(\mathbf{x}, \mathbf{x}_{s,j})$ is defined in Eq. 1. Therefore, the PLDA score space is defined by target-speaker's i-vectors through the PLDA model. Because H_s is typically small, the dimension of $\overrightarrow{S}_{\text{LR}}(\mathbf{x}, \mathcal{X}_s)$ is low.

3. PLDA-SVM with Empirical Kernel Maps

Support vector machine [20] (SVMs) are well-known supervised learning method used for classification and regression. Assume that we are given

[1] It is not necessary to apply utterance partitioning to non-target speakers because non-target i-vectors are abundant.

N training vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with labels $y_n \in \{+1, -1\}, n = 1, \ldots, N$. Using the pairs $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$, an SVM can be trained [20]. Given a test vector $\mathbf{x}_t$, the SVM's output is written as

$$f(\mathbf{x}_t; \mathbf{w}) = \sum_{i=1}^{N} w_i K(\mathbf{x}_t, \mathbf{x}_i) + w_0 \tag{6}$$

where $\mathbf{w} = [w_0, \ldots, w_N]$ are the weights determined by minimizing the error on the training set while maximizing the margin between the two classes, w_0 is a bias term, and $K(\mathbf{x}_t, \mathbf{x}_i)$ is a kernel function. This paper uses PLDA score vectors (via the empirical kernel maps) as the input to the SVMs and applies the speaker-dependent SVMs for i-vector/PLDA speaker verification. Specifically, Eq. 6 is rewritten as:

$$S_{\text{SVM}}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = \sum_{j \in \mathcal{S}_s} \alpha_{s,j} K(\mathbf{x}_t, \mathbf{x}_{s,j}) - \sum_{i \in \mathcal{S}_b} \alpha_{b,i} K(\mathbf{x}_t, \mathbf{x}_{b,i}) + w_0 \tag{7}$$

where $\mathcal{S}_s$ and $\mathcal{S}_b$ contain the indexes of the support vectors corresponding to the speaker class and impostor class, respectively. $\alpha_{s,j}$ and $\alpha_{b,i}$ are the Lagrange multipliers of the SVM. The relationship between $\mathbf{w}$ and α can be expressed as $w_n = \alpha_n y_n$. $K(\mathbf{x}_t, \mathbf{x}_{s,j})$ is a kernel function with the form:

$$K(\mathbf{x}_t, \mathbf{x}_{s,j}) = \mathbb{K}\left(\overrightarrow{S}_{\text{LR}}(\mathbf{x}_t, \mathcal{X}_s), \overrightarrow{S}_{\text{LR}}(\mathbf{x}_{s,j}, \mathcal{X}_s)\right) \tag{8}$$

where $\mathbb{K}(\cdot, \cdot)$ is a standard SVM kernel, e.g., linear or RBF. Only RBF kernel $\mathbb{K}(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\gamma^2})$ was adopted in this paper. $K(\mathbf{x}_t, \mathbf{x}_{b,i})$ can be obtained by replacing $\mathbf{x}_{s,j}$ in Eq. 8 with $\mathbf{x}_{b,i}$.

While our earlier studies [6, 7] have demonstrated that PLDA-SVM scoring (Eq. 7) performs better than simple PLDA scoring (Eq. 1), the SVMs in Eq. 7 still has some limitations [16].

1. Although SVM is a sparse model, the number of support vectors increases linearly with the size of the training set. In our case, this property limits the value of B (Eq. 4) for training the SVMs.

2. The SVM scores in Eq. 7 are not probabilistic, meaning that score normalization may be needed to adjust the score range of individual SVMs.

3. To achieve the best performance, it is necessary to strike a compromise between the training error and the margin of separation through ad-

justing the penalty factor for each target speaker during SVM training. Given the limited number of enrollment utterances for some speakers, this is not easy to achieve.

4. PLDA-RVM with Empirical Kernel Maps

To overcome the first and third limitations of SVMs mentioned in Section 3, this paper proposes incorporating the empirical kernel maps into another sparse kernel machine known as the relevance vector machine (RVM) [16], which leads to the PLDA-RVM scoring. To overcome the first limitation of SVM, PLDA-RVM makes use of the property of RVM to ensure that the number of relevant vectors does not grows linearly with the number of training vectors. To overcome the third limitation, PLDA-RVM takes the advantage of RVM by noting that there is no penalty factor in RVM training. As a result, only one hyper-parameter (the RBF width) needs to be adjusted.

In terms of output scoring, RVM [16] and SVM have the same form (Eq. 6). However, their learning methods are very different. The training of SVMs is based on structural risk minimization [21], whereas RVM training is based on Bayesian relevance learning [16] so that it provides a Bayesian treatment of Eq. 6. RVMs have two modes of operations: regression and classification. They are elaborated in the following subsections.

4.1. RVM Regression

Assume that for target speaker s, we have a set of training i-vectors $\mathcal{X} = \{\mathcal{X}_s, \mathcal{X}_b\}$ as in Eq. 3 and Eq. 4 and that $y_n = 1$ when $\mathbf{x}_n \in \mathcal{X}_s$ and $y_n = -1$ when $\mathbf{x}_n \in \mathcal{X}_b$. When an RVM is applied to regression, the targets y_n's are assumed to be sampled from the following model:[2]

$$y_n = f(\mathbf{x}_n; \mathbf{w}) + \epsilon_n, \quad n = 1, \dots, N,$$

where $N = |\mathcal{X}_s| + |\mathcal{X}_b|$, $f(\mathbf{x}_n; \mathbf{w})$ is given by Eq. 6, and ϵ_n follows a Gaussian distribution with zero mean and variance σ^2. This is equivalent to say that $p(y_n|\mathbf{x}_n) = \mathcal{N}(y_n|f(\mathbf{x}_n; \mathbf{w}), \sigma^2)$.

[2] To simplify notations in subsequence equations, we drop the subscripts s and b that annotate the target speaker and background speakers, respectively.

Assume also that y_n's $(n = 1, \ldots, N)$ are independent, the likelihood of the training data set can be written as:

$$\begin{aligned} p(\mathbf{y}|\mathbf{w}, \sigma^2) &= (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|^2\right\} \\ &= \mathcal{N}\left(\mathbf{y}|\boldsymbol{\Phi}\mathbf{w}, \sigma^2\mathbf{I}\right) \end{aligned} \tag{9}$$

where

$$\begin{aligned} \mathbf{y} = [y_1, \ldots, y_N]^{\mathsf{T}}; &\quad \mathbf{w} = [w_0, \ldots, w_N]^{\mathsf{T}}; \\ \boldsymbol{\Phi} &= [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)]^{\mathsf{T}} \\ \boldsymbol{\phi}(\mathbf{x}_i) &= [1, K(\mathbf{x}_i, \mathbf{x}_1), K(\mathbf{x}_i, \mathbf{x}_2), \ldots, K(\mathbf{x}_i, \mathbf{x}_N)]^{\mathsf{T}}. \end{aligned} \tag{10}$$

To avoid over-fitting, RVM defines a zero-mean Gaussian prior distribution over $\mathbf{w}$:

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=0}^{N} \mathcal{N}(w_i|0, \alpha_i^{-1}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}^{-1}) \tag{11}$$

where $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_N]^{\mathsf{T}}$, α_i is the hyperparameter associated with weight w_i and $\mathbf{A} = \mathrm{diag}(\alpha_0, \alpha_1, \ldots, \alpha_N)$.

Given the distribution of $\mathbf{y}$ in Eq. 9 and the prior distribution of $\mathbf{w}$ in Eq. 11, we can use the formula of conditional Gaussians (Eq. 2.116 in [22]) to obtain the posterior distribution over the weights as follows:[3]

$$p(\mathbf{w}|\mathbf{y}, \mathcal{X}, \boldsymbol{\alpha}, \sigma^2) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{12}$$

where

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathsf{T}}\mathbf{y} \quad \text{and} \quad \boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} + \mathbf{A})^{-1}. \tag{13}$$

The optimal value of $\boldsymbol{\alpha}$ and σ^2 can be obtained by maximizing the following marginal likelihood with respect to $\boldsymbol{\alpha}$ and σ^2:

$$\begin{aligned} p(\mathbf{y}|\mathcal{X}, \boldsymbol{\alpha}, \sigma^2) &= \int p(\mathbf{y}|\mathcal{X}, \mathbf{w}, \sigma^2) p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w} \\ &= \int \mathcal{N}(\mathbf{y}|\boldsymbol{\Phi}\mathbf{w}, \sigma^2\mathbf{I}) \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}^{-1}) d\mathbf{w} \\ &= \mathcal{N}(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathsf{T}}). \end{aligned} \tag{14}$$

[3] To use Eq. 2.116 of [22], we consider $\mathbf{x}$ and $\mathbf{y}$ in Eq. 2.116 as our $\mathbf{w}$ and $\mathbf{y}$, respectively. Also, $\boldsymbol{\Lambda}$ and $\mathbf{L}$ in Eq. 2.113 and Eq. 2.114 are our $\mathbf{A}$ and $\sigma^{-2}\mathbf{I}$, respectively. Moreover, $\boldsymbol{\mu}$ and $\mathbf{b}$ in Eq. 2.113 and Eq. 2.114 are zero vectors in our case.

Setting

$$\frac{\partial \ln p(\mathbf{y}|\mathcal{X}, \boldsymbol{\alpha}, \sigma^2)}{\partial \alpha_i} = 0 \quad \text{and} \quad \frac{\partial \ln p(\mathbf{y}|\mathcal{X}, \boldsymbol{\alpha}, \sigma^2)}{\partial \sigma^{-2}} = 0,$$

we obtain the following update formulae for α_i and σ^2 (see Appendix for the derivations):

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{\mu_i^2} \quad \text{and} \quad (\sigma^2)^{\text{new}} = \frac{\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2}{N - \sum_{i=0}^{N} \gamma_i} \tag{15}$$

where μ_i is the i-th component of $\boldsymbol{\mu}$ in Eq. 13 and $\gamma_i = 1 - \alpha_i \Sigma_{ii}$ with Σ_{ii} being the i-th diagonal element of $\boldsymbol{\Sigma}$ in Eq. 13. During the optimization, many of the hyperparameters α_i tend to infinity and the corresponding weights w_i become zero; the vectors $\mathbf{x}_i$ corresponding to the non-zero weights are considered as **relevance vectors**.

By considering $\mathbf{w}$ probabilistic and using the notion of conditional independence [22], the predictive distribution of y_t given a test vector $\mathbf{x}_t$ is

$$p(y_t|\mathbf{y}, \mathbf{x}_t, \mathcal{X}) = \int_{\sigma^2} \int_{\boldsymbol{\alpha}} \int_{\mathbf{w}} p(y_t|\mathbf{x}_t, \mathbf{w}, \boldsymbol{\alpha}, \sigma^2) p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{y}, \mathcal{X}) d\mathbf{w} d\boldsymbol{\alpha} d\sigma^2 \tag{16}$$

where

$$p(y_t|\mathbf{x}_t, \mathbf{w}, \boldsymbol{\alpha}, \sigma^2) = p(y_t|\mathbf{x}_t, \mathbf{w}, \sigma^2) \tag{17}$$

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{y}, \mathcal{X}) = p(\mathbf{w}|\mathbf{y}, \mathcal{X}, \boldsymbol{\alpha}, \sigma^2) p(\boldsymbol{\alpha}, \sigma^2|\mathbf{y}, \mathcal{X}). \tag{18}$$

Instead of computing the posterior $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{y})$ in Eq. 17, Tipping [16] used a delta function at the most probable values of $\boldsymbol{\alpha}$ and σ^2 as an approximation. Therefore, using Eq. 18 and assuming uniform priors for $\boldsymbol{\alpha}$ and σ^2, Eq. 16 reduces to

$$\begin{aligned} p(y_t|\mathbf{y}, \mathbf{x}_t, \mathcal{X}) &= \int_{\sigma^2} \int_{\boldsymbol{\alpha}} \int_{\mathbf{w}} p(y_t|\mathbf{x}_t, \mathbf{w}, \boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}}) p(\mathbf{w}|\mathbf{y}, \mathcal{X}, \boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}}) p(\boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}}|\mathbf{y}) d\mathbf{w} d\boldsymbol{\alpha} d\sigma^2 \\ &= \int_{\mathbf{w}} p(y_t|\mathbf{x}_t, \mathbf{w}, \boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}}) p(\mathbf{w}|\mathbf{y}, \mathcal{X}, \boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}}) d\mathbf{w} \\ &= \int_{\mathbf{w}} \mathcal{N}(y_t|\boldsymbol{\phi}(\mathbf{x}_t)^{\mathsf{T}}\mathbf{w}, \sigma^2_{\text{MP}}) \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_{\text{MP}}, \boldsymbol{\Sigma}_{\text{MP}}) d\mathbf{w} \end{aligned} \tag{19}$$

where

$$
\begin{aligned}
(\boldsymbol{\alpha}_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}}) &= \arg\max_{\boldsymbol{\alpha},\sigma^2} p(\boldsymbol{\alpha}, \sigma^2|\mathbf{y}, \mathcal{X}) \\
&= \arg\max_{\boldsymbol{\alpha},\sigma^2} p(\mathbf{y}|\boldsymbol{\alpha}, \sigma^2, \mathcal{X})p(\boldsymbol{\alpha})p(\sigma^2) \\
&= \arg\max_{\boldsymbol{\alpha},\sigma^2} \int p(\mathbf{y}|\mathbf{w}, \sigma^2, \mathcal{X})p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w} \\
&= \arg\max_{\boldsymbol{\alpha},\sigma^2} \int \mathcal{N}(\mathbf{y}|\boldsymbol{\Phi}\mathbf{w}, \sigma^2\mathbf{I})\mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{A}^{-1})d\mathbf{w} \\
&= \arg\max_{\boldsymbol{\alpha},\sigma^2} \mathcal{N}(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathsf{T}}). \qquad (20)
\end{aligned}
$$

and

$$\boldsymbol{\mu}_{\mathrm{MP}} = \sigma^{-2}_{\mathrm{MP}}\boldsymbol{\Sigma}_{\mathrm{MP}}\boldsymbol{\Phi}^{\mathsf{T}}\mathbf{y} \quad \text{and} \quad \boldsymbol{\Sigma}_{\mathrm{MP}} = \left(\sigma^{-2}_{\mathrm{MP}}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} + \mathbf{A}\right)^{-1}. \qquad (21)$$

Because both terms in the integrand of Eq. 19 are Gaussians, the predictive distribution is also a Gaussian:[4]

$$p(y_t|\mathbf{y}, \mathcal{X}, \mathbf{x}_t, \boldsymbol{\alpha}_{\mathrm{MP}}, \sigma^2_{\mathrm{MP}}) = \mathcal{N}(y_t|g(\mathbf{x}_t), \sigma^2_t) \qquad (22)$$

with

$$g(\mathbf{x}_t) = \boldsymbol{\mu}^{\mathsf{T}}_{\mathrm{MP}}\boldsymbol{\phi}(\mathbf{x}_t) \quad \text{and} \quad \sigma^2_t = \sigma^2_{\mathrm{MP}} + \boldsymbol{\phi}(\mathbf{x}_t)^{\mathsf{T}}\boldsymbol{\Sigma}_{\mathrm{MP}}\boldsymbol{\phi}(\mathbf{x}_t). \qquad (23)$$

To use RVM regression for PLDA-based speaker verification, we train one RVM regressor for each target speaker, i.e., each speaker has his/her own $\boldsymbol{\mu}_{\mathrm{MP}}$ in Eq. 21. During verification, given a test i-vector $\mathbf{x}_t$ and a target speaker s, we use the posterior mean $g(\mathbf{x}_t)$ in Eq. 23 as the verification score. More specifically,

$$S_{\text{RVM-R}}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = g(\mathbf{x}_t) = \boldsymbol{\mu}^{\mathsf{T}}_{\mathrm{MP},s}\boldsymbol{\phi}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) \qquad (24)$$

where

$$\boldsymbol{\phi}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = [1, K(\mathbf{x}_t, \mathbf{x}_{s,1}), \ldots, K(\mathbf{x}_t, \mathbf{x}_{s,H_s}), K(\mathbf{x}_t, \mathbf{x}_{b,1}), \ldots, K(\mathbf{x}_t, \mathbf{x}_{b,B})]^{\mathsf{T}} \qquad (25)$$

[4] Again, we make use of the marginal and conditional Guassian formulae in Eqs. 2.113–2.115 of [22] to derive Eq. 23. Specifically, in Eqs. 2.113–2.115 of [22], we substitute $\mathbf{x}$ by $\mathbf{w}$, $\mathbf{y}$ by y_t, $\boldsymbol{\mu}$ by $\boldsymbol{\mu}_{\mathrm{MP}}$, $\boldsymbol{\Lambda}^{-1}$ by $\boldsymbol{\Sigma}_{\mathrm{MP}}$, $\mathbf{L}^{-1}$ by σ^2_{MP}, $\mathbf{b}$ by 0, and $\mathbf{A}$ by $\phi(\mathbf{x})^{\mathsf{T}}$.

4.2. RVM Classification with Empirical Kernel Maps

When RVM is applied to classification, the target conditional distribution $p(y|\mathbf{x})$ is assumed to follow a Bernoulli distribution. Assume a set of training i-vectors $\mathcal{X} = \{\mathcal{X}_s, \mathcal{X}_b\}$ as in Eq. 3 and Eq. 4 and $y_i = 1$ when $\mathbf{x}_i \in \mathcal{X}_s$ and $y_i = 0$ when $\mathbf{x}_i \in \mathcal{X}_b$ for target speaker s, the likelihood of the training data set can be written as [16]:

$$p(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{N} \sigma\left(f(\mathbf{x}_i;\mathbf{w})\right)^{y_i} \left\{1 - \sigma\left(f(\mathbf{x}_i;\mathbf{w})\right)\right\}^{1-y_i}, \quad y_i \in \{0,1\}. \tag{26}$$

where

$$N = |\mathcal{X}_s| + |\mathcal{X}_b|;\ \mathbf{y} = [y_1, \ldots, y_N]^{\mathsf{T}}; \quad \mathbf{w} = [w_0, \ldots, w_N]^{\mathsf{T}} \tag{27}$$

and $\sigma\{\cdot\}$ is the logistic sigmoid link function $\sigma(z) = \frac{1}{1+e^{-z}}$. Similar to RVM regression, RVM classification also introduces a zero-mean Gaussian prior distribution over $\mathbf{w}$ as defined in Eq. 11.

Using Eq. 26 and Eq. 11, we can obtain the posterior distribution of $\mathbf{w}$:

$$p(\mathbf{w}|\mathbf{y},\boldsymbol{\alpha}) = \frac{p(\mathbf{y}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})}{\int p(\mathbf{y}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w}} = \frac{g(\mathbf{w})}{p(\mathbf{y}|\boldsymbol{\alpha})}, \tag{28}$$

where we have defined $g(\mathbf{w}) \equiv p(\mathbf{y}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})$. Taking logarithm of $g(\mathbf{w})$, we have

$$\begin{aligned}\log g(\mathbf{w}) &= \log p(\mathbf{y}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha}) \\ &= \sum_{i=1}^{N} \left\{y_i \log\left[\sigma(f(\mathbf{x}_i;\mathbf{w}))\right] + (1-y_i)\log\left[1-\sigma(f(\mathbf{x}_i;\mathbf{w}))\right]\right\} \\ &\qquad\quad - \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{A}\mathbf{w} + \text{const} \\ &= \sum_{i=1}^{N} \left\{y_i \log\left[\sigma\left(\boldsymbol{\phi}(\mathbf{x}_i)^{\mathsf{T}}\mathbf{w}\right)\right] + (1-y_i)\log\left[1-\sigma\left(\boldsymbol{\phi}(\mathbf{x}_i)^{\mathsf{T}}\mathbf{w}\right)\right]\right\} \\ &\qquad\quad - \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{A}\mathbf{w} + \text{const},\end{aligned} \tag{29}$$

where we have used $f(\mathbf{x}_i;\mathbf{w}) = \boldsymbol{\phi}(\mathbf{x}_i)^{\mathsf{T}}\mathbf{w}$. Note that because $p(\mathbf{y}|\mathbf{w})$ in Eq. 26 is not a Gaussian, we cannot analytically perform the integration in Eq. 28 to obtain a closed-form solution for $p(\mathbf{w}|\mathbf{y},\boldsymbol{\alpha})$. One possible solution is to use the Laplace's method [22] to approximate $p(\mathbf{w}|\mathbf{y},\boldsymbol{\alpha})$ by a Gaussian distribution. The idea is to find a Gaussian approximation $q(\mathbf{w})$ with mean $\mathbf{w}_0$ equals to a mode of $p(\mathbf{w}|\mathbf{y},\boldsymbol{\alpha})$. This can be achieved by approximating $\log g(\mathbf{w})$ to a Taylor expansion around $\mathbf{w}_0$:

$$\log g(\mathbf{w}) \approx \log g(\mathbf{w}_0) - \frac{1}{2}\left(\mathbf{w}-\mathbf{w}_0\right)^{\mathsf{T}} \mathbf{H} \left(\mathbf{w}-\mathbf{w}_0\right), \tag{30}$$

where $\mathbf{H}$ is a Hessian matrix

$$\begin{aligned}
\mathbf{H} &= -\nabla\nabla_{\mathbf{w}} \log g(\mathbf{w})\Big|_{\mathbf{w}=\mathbf{w}_0} & (31)\\
&= \frac{\partial}{\partial\mathbf{w}\partial\mathbf{w}^{\mathsf{T}}} \log g(\mathbf{w})\Big|_{\mathbf{w}=\mathbf{w}_0} \\
&= \sum_{i=1}^{N} \sigma\left(\phi(\mathbf{x}_i)^{\mathsf{T}}\mathbf{w}_0\right)\left[1-\sigma\left(\phi(\mathbf{x}_i)^{\mathsf{T}}\mathbf{w}_0\right)\right]\phi(\mathbf{x}_i)^{\mathsf{T}}\phi(\mathbf{x}_i) + \mathbf{A} \\
&= \mathbf{\Phi}^{\mathsf{T}}\mathbf{B}\mathbf{\Phi} + \mathbf{A} & (32)
\end{aligned}$$

where $\mathbf{B}$ is an $(N+1)\times(N+1)$ diagonal matrix with diagonal elements

$$b_{ii} = \sigma\left(\phi(\mathbf{x}_i)^{\mathsf{T}}\mathbf{w}_0\right)\left[1-\sigma\left(\phi(\mathbf{x}_i)^{\mathsf{T}}\mathbf{w}_0\right)\right], \tag{33}$$

and $\mathbf{\Phi}$ and $\boldsymbol{\phi}(\mathbf{x}_t)$ are defined in Eq. 10.

The value of $\mathbf{w}_0$ can be obtained by using iterative reweighted least squares (IRLS) [22] as follows:

$$\mathbf{w}_0^{\text{new}} = \mathbf{w}_0^{\text{old}} - (\mathbf{H}^{\text{old}})^{-1}\nabla_{\mathbf{w}} \log g(\mathbf{w})\Big|_{\mathbf{w}=\mathbf{w}_0^{\text{old}}} \tag{34}$$

where

$$\nabla_{\mathbf{w}} \log g(\mathbf{w}) = \mathbf{\Phi}^{\mathsf{T}}\left(\mathbf{y} - \left[\sigma(\phi(\mathbf{x}_1)^{\mathsf{T}}\mathbf{w}),\ldots,\sigma(\phi(\mathbf{x}_N)^{\mathsf{T}}\mathbf{w})\right]^{\mathsf{T}}\right) - \mathbf{A}\mathbf{w}.$$

At convergency, the gradient is zero and therefore we have

$$\mathbf{w_0} \to \mathbf{A}^{-1}\mathbf{\Phi}^{\mathsf{T}}\left(\mathbf{y} - \left[\sigma(\phi(\mathbf{x}_1)^{\mathsf{T}}\mathbf{w}_0),\ldots,\sigma(\phi(\mathbf{x}_N)^{\mathsf{T}}\mathbf{w}_0)\right]^{\mathsf{T}}\right)$$

Taking exponential of Eq. 30 and noting that $q(\mathbf{w}) \propto g(\mathbf{w})$, we have

$$\begin{aligned} q(\mathbf{w}) &= \frac{|\mathbf{H}|^{1/2}}{(2\pi)^{(N+1)/2}} \exp\left\{ -\frac{1}{2} \left(\mathbf{w} - \mathbf{w}_0\right)^{\mathsf{T}} \mathbf{H} \left(\mathbf{w} - \mathbf{w}_0\right) \right\} \\ &= \mathcal{N}(\mathbf{w}|\mathbf{w}_0, \mathbf{H}^{-1}), \end{aligned} \tag{35}$$

which is a Gaussian distribution with mean $\mathbf{w}_0$ and covariance matrix $\mathbf{H}^{-1}$.

We then use $q(\mathbf{w})$ to approximate the posterior $p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha})$ around the mode $\mathbf{w}_0$. Comparing the covariance matrix $\mathbf{H}^{-1}$ in Eq. 31 with that in Eq. 2.117 of [22] reveals that $\mathbf{B}$ is the precision matrix of $p(\mathbf{y}|\mathbf{w})$ (see Eq. 2.114 of [22]). As a result, using Eq. 2.116 of [22], we obtain the posterior mean of the weights in $p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha})$ as

$$\mathbf{w}_{\text{MP}} = \mathbf{H}^{-1}\boldsymbol{\Phi}^{\mathsf{T}}\mathbf{B}\mathbf{y}. \tag{36}$$

Given Eqs. 15, 31, 33, 34, and 36, we may proceed the estimation of $\boldsymbol{\alpha}$ as follows. First, we initialize $\boldsymbol{\alpha}$ to obtain $\mathbf{A}$. Then, we initialize $\mathbf{w}$ and use Eq. 34 to estimate $\mathbf{w}_0$. We then plug this $\mathbf{w_0}$ into Eq. 31 and Eq. 33 to obtain $\mathbf{H}$ and $\mathbf{B}$, respectively, followed by estimating $\mathbf{w}_{\text{MP}}$ using Eq. 36. A new estimation of $\boldsymbol{\alpha}$ is then obtained by maximizing the likelihood $p(\mathbf{y}|\boldsymbol{\alpha})$, i.e., using Eq. 15 without σ^2. Then, the cycle is repeated.

To apply RVM classification for PLDA-based speaker verification, we train one RVM classifier for each speaker, i.e., each speaker his/her own $\mathbf{w}_{\text{MP}}$ in Eq. 36. During verification, given a test i-vector $\mathbf{x}_t$ and a target speaker s, we use $\sigma(\mathbf{w}_{\text{MP},s}^{\mathsf{T}}\phi(\mathbf{x}_t))$ as the score. More precisely,

$$S_{\text{RVM-C}}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = \sigma\left(\mathbf{w}_{\text{MP},s}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b)\right) \tag{37}$$

where

$$\boldsymbol{\phi}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = [1, K(\mathbf{x}_t, \mathbf{x}_{s,1}), \ldots, K(\mathbf{x}_t, \mathbf{x}_{s,H_s}), K(\mathbf{x}_t, \mathbf{x}_{b,1}), \ldots, K(\mathbf{x}_t, \mathbf{x}_{b,B})]^{\mathsf{T}} \tag{38}$$

and $K(\mathbf{x}_i, \mathbf{x}_j)$ is defined in Eq. 8. For the convenience of plotting DET curves, this paper uses the linear function ($\sigma(z) = z$) instead of logistic sigmoid link function in Eq. 37.

5. Experimental Setup

5.1. Speech Data and Acoustic Features

The *core set* of NIST 2012 Speaker Recognition Evaluation (SRE) [15] was used for performance evaluation. This paper focuses on the phonecall speech of the core task, i.e., Common Evaluation Conditions 2, 4, and 5. Hereafter, we use “CC” to denote common evaluation conditions. In the evaluation dataset, no noise was added to the test segments of CC2, whereas noise was added to the test segments of CC4 and test segments in CC5 were collected in a noisy environment. All of these conditions contain training segments with variable length and variable numbers of training segments per target speaker. We removed the 10-second utterances and the summed-channel utterances from the training segments of NIST 2012 SRE but ensured that all target speakers have at least one long utterance for enrollment. The speech files in NIST 2005–2010 SREs were used as development data for training the UBM, total variability matrix, LDA-WCCN, PLDA models, SVMs and RVMs.

We used our voice activity detector [23, 24] to detect the speech regions of each utterance. 19 MFCCs together with energy plus their 1st- and 2nd- derivatives were extracted from the speech regions, followed by cepstral mean normalization [25] and feature warping [26] with a window size of 3 seconds. A 60-dim acoustic vector was extracted every 10ms, using a Hamming window of 25ms.

To improve noise robustness, we followed the suggestions in [27] to add noise to the training files. To this end, we constructed a noise dataset comprising 13 real crowd noise files and 17 heating, ventilation, and air conditioning (HVAC) noise files from [28] and 10 artificial crowd noise files generated by summing 441 utterances from male and female speakers in pre-2012 NIST SRE. For each training file with SNR above 15dB, we generated two noisy speech files at an SNR of 6dB and 15dB by randomly selecting two noise files from the noise dataset. For each training file with SNR between 6dB and 15dB, we produced a noisy speech file at 6dB. The SNRs of test files were estimated by the speech voltmeter function in FaNT [29] and the VAD decisions. Specifically, we used the VAD [23, 24] to determine speech and non-speech regions in a speech file. Then, energies of the speech and non-speech parts as determined by the voltmeter function of FaNT were used for estimating the SNR.

5.2. Enrollment Utterances and Non-Target Speaker Utterances

Because the test conditions involve phonecall speech only, only telephone utterances were selected as enrollment utterances for matching the channel between enrollment and test sessions. Although many target speakers in NIST 2012 SRE have multiple training segments, some of them have a few training segments only. More precisely, after removing the 10-second segments and summed-channel segments, 50 out of 723 male target speakers and 65 out of 1,095 female target speakers have one long training segment only. To provide more speaker-class i-vectors for creating the empirical kernel maps and for training the SVMs/RVMs for these speakers, we used a technique called utterance partitioning with acoustic vector resampling (UP-AVR) [30, 19]. Specifically, for each conversation, a sequence of acoustic vectors is extracted. Then, the sequence is partitioned into N equal-length segments, and an i-vector is estimated from each segment. If more i-vectors are required, the acoustic vectors in the sequence are randomly reshuffled and the partitioning process is repeated to produce another N vectors. If this partitioning-randomization process is repeated R times, $(RN + 1)$ i-vectors can be obtained from a single conversation, where the additional one is obtained from the entire acoustic sequence.

Our earlier studies [31, 30, 19] suggest that to facilitate the SVM training algorithm to find a good decision boundary, it is necessary to minimize the imbalance between the numbers of speaker-class and impostor-class training vectors. To this end, we propose the following rule to produce H_s speaker-class training i-vectors for speaker s:

$$H_s = \begin{cases} 17, & |\mathcal{U}_s| < 17 \\ |\mathcal{U}_s|, & |\mathcal{U}_s| \geq 17 \end{cases} \tag{39}$$

where $\mathcal{U}_s$ is the set of full-length enrollment utterances and $|\mathcal{U}_s|$ represents its cardinality. To make the full use of the enrollment utterances of a target speaker, UP-AVR ($N = 4$ and $R = 4$) was performed on each full-length enrollment utterance to produce a sub-utterance set, one for each utterance. Then we selected sub-utterances from these sets uniformly to make the total number of enrollment utterances equal to 17. For example, if $|\mathcal{U}_s|$ of a target speaker is equal to 5, 5 sub-utterances sets are produced from these 5 full-length enrollment utterances by UP-AVR ($N = 4$ and $R = 4$). Each sub-utterance set contains 16 sub-utterances. To fully use the enrollment utterances of this target speaker, 3 sub-utterances were extracted from each of two sub-utterances sets and 2 sub-utterances were extracted from each of other three sub-utterances sets. Eventually, we have 12 sub-utterances and

5 full-length utterances for this target speaker.

We adopted known non-targets [7] to train SVMs and RVMs in this paper. For each target speaker, 500 competing known non-target speakers were randomly selected and each known non-target speaker provides one utterance. Therefore, 500 utterances were used to train his/her SVM/RVM for all common conditions. Note that RVM classification applies a logistic link function (Eq. 37) to compute the probabilistic outputs (posterior probabilities of the target-speaker class) [16]. While probabilistic outputs are desirable when the classification task involves one RVM only, in NIST SRE, we have one RVM per target speaker and the performance indexes (EER, minDCF, and DET) are based on the scores of all true-speaker trials and impostor attempts. This will lead to two skewed score-distributions with modes close to 1 and 0 for true-speaker trials and impostor attempts, respectively. Although these skewed distribution do not hurt the performance of SRE, we only apply the logistic sigmoid function during the training of RVM classifiers and dropped the function during scoring so that the score distribution of RVM classification is consistent with that of other methods. More precisely, Eq. 6 was used for computing the verification scores in the classification mode of RVMs and SVMs in our experiments.

5.3. Total Variability Modeling and PLDA

The i-vector systems are based on a gender-dependent UBM with 1024 mixtures. 3,500 microphone utterances and 3,501 telephone utterances from NIST 2005–2008 SREs were used for training the male UBM. 4,177 microphone utterances and 4,178 telephone utterances from NIST 2005–2008 SREs were used for training the female UBM. We selected 14,875 telephone and interview conversations from 575 speakers in NIST 2006–2010 SREs to estimate male total variability matrix with 400 total factors and 20,656 telephone and interview conversations from 889 speakers in NIST 2006–2010 SREs to estimate female total variability matrix with 400 total factors.

According to [32], adding noisy data to train the UBM and total variability matrix receives very small gains. Hence, we followed the steps in [32] and only applied noisy data to train the LDA and PLDA parameters. For the common condition without added noise (CC2), we used clean utterances from NIST 2006–2010 SREs to train the PLDA models. For the common conditions with added noise (CC4 and CC5), we pooled clean utterances, utterances at 6 dB SNR, and utterances at 15 dB SNR to estimate the loading matrix.

We used within-class covariance normalization (WCCN) [33] for whitening [34] the i-vectors, followed by vector-length normalization [3]. Then, we performed linear discriminant analysis (LDA) [22] and WCCN on the resulting vectors to reduce the dimension to 200 before training the PLDA models with 150 latent variables.

6. Results and Discussions

6.1. Property of Empirical Kernel Maps in SVM and RVM

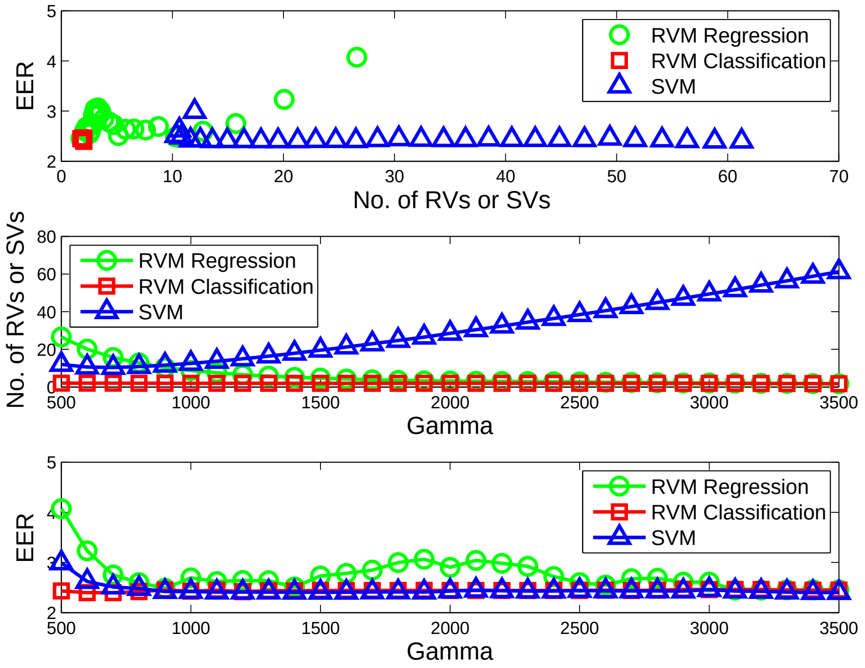


Figure 1: The property of empirical kernel maps in SVMs, RVM regressions, and RVM classifications. Gamma is the RBF parameter γ.

It is of interest to investigate the property of empirical kernel maps in SVM and RVM. To this end, 108 target speakers with true-target trials and imposter trials were extracted from CC2 of NIST 2012 SRE. Using these trials, the equal error rates (EERs) achieved by the SVMs and RVMs and their corresponding number of support vectors (SVs) and relevant vectors (RVs) were averaged across the 108 speakers. An RBF kernel was adopted, where the RBF parameters γ was varied from 500 to 2500.[5] The experiments

[5] Because the LR scores have range between −579.5 to 199.8 and the dimension of the

on RVM were based on the Matlab code from Tipping [35]. The penalty factor C was set to 1 for all SVMs.

The top panel of Figure 1 plots the average EER against the average number of SVs and RVs in the SVMs and RVMs. It clearly shows that the performance of SVMs is fairly stable with respect to the number of SVs. On the other hand, when the number of RVs increases, the performance of RVM regression becomes poor. In addition, even though the number of RVs in RVM classification is very small, its performance is comparable with that of SVM classification.

The middle panel of Figure 1 shows that when the RBF parameter γ increases, the number of SVs gradually increases. On the other hand, the number of RVs in RVM regression monotonically decreases when γ increases. More importantly, for a wide range of γ, there are more SVs than RVs, suggesting that for this dataset, both RVM classification and RVM regression are sparser than the SVMs.

The bottom panel of Figure 1 demonstrates that the performance of both SVM and RVM classification are stable with respect to γ, while RVM regression is more sensitive to the value of γ.

We have also investigated the effect of increasing the number of training score-vectors on SVM and RVM. To this end, 723 male target speakers were selected from NIST 2012 SRE. Each of these speakers has at least 17 enrollment utterances for constructing the empirical kernel maps (score vectors) and for training their SVM/RVM. Because the number of speaker-class training samples is small, we only varied the number of imposter-class training samples from 100 to 700. An RBF kernel was adopted, where the RBF parameter γ was fixed to 1500 for both SVM and RVM training. Figure 2 shows the numbers of relevance vectors (RVs) and support vectors (SVs) with respect to the number of imposter-class training vectors, where these numbers are based on the average of 723 target speakers. The figure shows that the number of support vectors grows linearly with the number of imposter-class training samples, whereas the number of relevance vectors is relatively stable. This result demonstrates the first limitation of SVM (cf. Section 3) and shows that RVM can overcome this limitation.

6.2. PLDA Scoring vs. Sparse Kernel Machines with EKMs

Table 1 compares the performance between conventional PLDA score averaging and sparse kernel machines with EKMs for male and female speakers

LDA-projected i-vector is 150, a large value of γ is necessary.

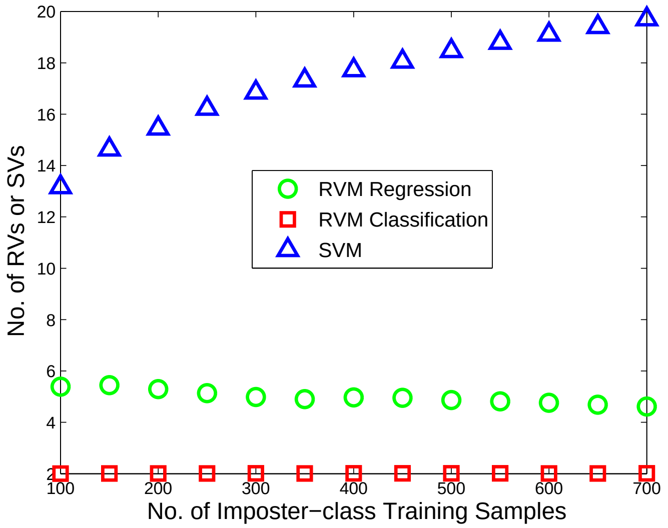


Figure 2: The numbers of support vectors (SVs) and relevance vectors (RVs) versus the number of imposter-class training samples.

in CC2, CC4, and CC5 of NIST 2012 SRE. Table 1 demonstrates that the performance of sparse kernel machines with EKMs is better than that of the conventional PLDA scoring. More specifically, in terms of CC2 (male speakers), SVM scoring with EKMs can reduce the EER of PLDA scoring from 2.40% to 1.84%, which amounts to 23.3% relative reduction. Similarly, the method reduces the minimum DCF from 0.333 to 0.306, which amounts to 8.1% relative reduction. For female speakers in CC2, SVM scoring with EKMs also can achieve around 4.3% and 7.8% relative reduction on EER and minimum DCF, respectively.

The scores of PLDA+UP-AVR+SVM and PLDA+UP-AVR+RVM-C were fused using a set of linear fusion weights that achieve the best fusion performance (in terms of minimum EER). Table 1 shows that the fusion improves the performance for CC2, CC4, and CC5. Moreover, Figure 3 shows the DET curves of the scoring methods. Both fusion performance and Figure 3 further demonstrate the benefits of sparse kernel machines with EKMs. The good performance of sparse kernel machines with EKMs is due to the fact that SVM and RVM not only utilize the information from the claimant and target speaker, but also take the background speaker information into consideration during the scoring process. In addition, the

Scoring Methods	EER (%)			MinNDCF(2012)		
	CC2	CC4	CC5	CC2	CC4	CC5
(A)PLDA	2.40	3.24	3.11	0.333	0.333	0.325
(B)PLDA+UP-AVR	2.43	3.21	3.09	0.327	0.334	0.328
(C)PLDA+UP-AVR+SVM	1.84	2.88	2.67	0.306	0.289	0.299
(D)PLDA+UP-AVR+RVM-C	2.15	3.12	2.66	0.303	0.293	0.290
(E)PLDA+UP-AVR+RVM-R	2.19	3.20	2.74	0.317	0.290	0.288
(C)+(D)	**1.80**	**2.85**	**2.53**	**0.301**	**0.283**	**0.287**

(a) Male

Scoring Methods	EER (%)			MinNDCF(2012)		
	CC2	CC4	CC5	CC2	CC4	CC5
(A)PLDA	2.08	2.59	2.77	0.348	0.332	0.342
(B)PLDA+UP-AVR	2.22	2.65	2.94	0.342	0.333	0.332
(C)PLDA+UP-AVR+SVM	1.99	2.48	2.70	0.321	0.342	0.325
(D)PLDA+UP-AVR+RVM-C	2.07	2.54	2.63	0.306	0.320	**0.316**
(E)PLDA+UP-AVR+RVM-R	2.12	2.59	2.68	**0.300**	**0.312**	0.321
(C)+(D)	**1.91**	**2.42**	**2.55**	0.314	0.325	0.318

(b) Female

Table 1: Performance of scoring methods in NIST 2012 SRE under the common conditions that involve telephone recordings. *"RVM-C"* represents relevance vector machine classification. *"RVM-R"* represents relevance vector machine regression. *"UP-AVR"* represents utterance partitioning with acoustic vector resampling [19]. The methods are named by the processes applied to the i-vectors for computing the verification scores. For example, *"PLDA+UP-AVR+SVM"* means that UP-AVR has been applied to create target-speaker i-vectors for constructing the EKMs and training SVMs. The RBF parameter γ for sparse kernel machines with EKMs was fixed to 1500.

contribution of individual background speakers and the target speaker can be optimally weighted by the SVM and RVM weights.

Because sparse kernel machines with EKMs require PLDA scores as input, their computation cost will be slightly higher than that of PLDA scoring. A method to reduce the computation cost is outlined in Appendix

B.

6.3. PLDA-RVM Scoring vs. PLDA-SVM Scoring

This section compares PLDA-RVM scoring with PLDA-SVM scoring from the following three perspectives:

1. **Evaluation performance**: Table 1 shows the performance of PLDA+UP-AVR+SVM, PLDA+UP-AVR+RVM-C, and PLDA+UP-AVR+RVM-R under different common conditions in NIST 2012 SRE and demonstrates that the performance of PLDA-RVM scoring is comparable with PLDA-SVM scoring. More specifically, for female speakers, the minimum DCF of PLDA-RVM scoring is better than that of PLDA-SVM scoring, but its EER is slightly worse than the EER of PLDA-SVM scoring; For male speakers, the performance of PLDA-RVM scoring is slightly worse than that of PLDA-SVM scoring, but it is still comparable. Figure 3 also shows that PLDA-RVM scoring achieve the similar performance as PLDA-SVM scoring.

2. **Sparsity**: Figure 4 compares the sparseness of the resulting PLDA-SVM, PLDA-RVM classification and PLDA-RVM regression models. The results indicate that the PLDA-RVM models are much sparser than PLDA-SVM model.

3. **Real scoring time**: Table 2 shows the scoring time[6] for each test trial under different scoring methods. It demonstrates that the scoring time of PLDA-RVM is less than PLDA-SVM. This is reasonable, because the number of RVs is much smaller than that of SVs, which agrees with the conclusion of Figure 4.

To sum up, PLDA-RVM models are much sparser than PLDA-SVM model; however, they achieve similar performance as PLDA-SVM.

7. Conclusions

This paper investigates the property of empirical kernel maps in SVM and RVM and compares the performance among these three classifiers in PLDA-based speaker verification. Experimental results show that PLDA-RVM is more sparse than PLDA-SVM, but it achieves comparable performance as PLDA-SVM. In addition, this paper also provides one way to speed

[6] The experiments were performed on an Intel Core Q9550 CPU.

Scoring Methods	Scoring Time (ms)
PLDA	0.427
PLDA+UP-AVR	0.431
PLDA+UP-AVR+SVM	0.695
PLDA+UP-AVR+RVM-C	0.524
PLDA+UP-AVR+RVM-R	0.528

Table 2: Scoring time for different scoring methods.

up the processing time of sparse kernel machines with EKMs. The idea of combining RVM with PLDA can be further explored in future work. For example, it is interesting to exploit the property that the kernel function used in RVM do not need to fulfill the Mercer's condition.

Appendix A: Estimating Hyperparameters

In Sections 3.5.1 and 3.5.2 of [22], the hyperparameters α is estimated by using the fact that the eigenvalues of $(\alpha\mathbf{I}+\mathbf{L})$ are $(\alpha+\lambda_i)$, where λ_i's are the eigenvalues of symmetric matrix $\mathbf{L}$. However, this property does not hold for $\mathbf{A}+\mathbf{L}$, where the diagonal elements of $\mathbf{A}$ are not equal, which is the case in RVM where $\mathbf{A}=\text{diag}\{\alpha_0,\ldots,\alpha_N\}$. More precisely, the eigenvalues of $(\mathbf{A}+\mathbf{L})$ are not equal to $\alpha_i+\lambda_i, i=0,\ldots,N$.

Instead of completing the square over $\mathbf{w}$, the optimal value of $\boldsymbol{\alpha}$ and σ^2 can be obtained by maximizing the following marginal likelihood with respect to $\boldsymbol{\alpha}$ and σ^2:

$$\begin{aligned} p(\mathbf{y}|\mathcal{X},\boldsymbol{\alpha},\sigma^2) &= \int p(\mathbf{y}|\mathcal{X},\mathbf{w},\sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w} \\ &= \int \mathcal{N}(\mathbf{y}|\boldsymbol{\Phi}\mathbf{w},\sigma^2\mathbf{I})\mathcal{N}(\mathbf{w}|\mathbf{0},\mathbf{A}^{-1})d\mathbf{w} \\ &= \mathcal{N}(\mathbf{y}|\mathbf{0},\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\mathsf{T}), \end{aligned} \tag{40}$$

where $\mathbf{A}=\text{dig}\{\alpha_0,\ldots,\alpha_N\}$. Taking natural logarithm of Eq. 40 and ignoring terms independent of $\boldsymbol{\alpha}$ and σ^2, the log-likelihood of $\mathbf{y}$ becomes

$$\mathcal{L}(\mathbf{y}|\boldsymbol{\alpha},\sigma^2) = -\frac{1}{2}\ln|\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\mathsf{T}| - \frac{1}{2}\mathbf{y}^\mathsf{T}(\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\mathsf{T})^{-1}\mathbf{y} \tag{41}$$

The first term can be expressed as

$$
\begin{aligned}
-\frac{1}{2}&\ln|\sigma^2\mathbf{I}+\mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^\mathsf{T}| \\
&=\frac{1}{2}\left(\ln|\mathbf{A}|-\ln|\sigma^2\mathbf{I}|-\ln|\mathbf{A}+\sigma^{-2}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi}|\right) \\
&=\frac{1}{2}\left(\sum_{i=0}^{N}\ln\alpha_i-N\ln\sigma^2+\ln|\mathbf{\Sigma}|\right)
\end{aligned}
\tag{42}
$$

where $\mathbf{\Sigma}=\left(\mathbf{A}+\sigma^{-2}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi}\right)^{-1}$ and we have used the determinant identity

$$
|\mathbf{A}||\sigma^2\mathbf{I}+\mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^\mathsf{T}|=|\sigma^2\mathbf{I}||\mathbf{A}+\sigma^{-2}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi}|.
$$

Using the Woodbury inversion identity, the second term in Eq. 41 can be expressed as

$$
\begin{aligned}
-\frac{1}{2}&\mathbf{y}^\mathsf{T}(\sigma^2\mathbf{I}+\mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^\mathsf{T})^{-1}\mathbf{y} \\
&=-\frac{1}{2}\mathbf{y}^\mathsf{T}\left[\sigma^{-2}\mathbf{I}-\sigma^{-2}\mathbf{\Phi}(\mathbf{A}+\sigma^{-2}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\mathsf{T}\sigma^{-2}\right]\mathbf{y} \\
&=-\frac{\sigma^{-2}}{2}\left[\mathbf{y}^\mathsf{T}\mathbf{y}-\mathbf{y}^\mathsf{T}\mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Phi}^\mathsf{T}\sigma^{-2}\mathbf{y}\right] \\
&=-\frac{1}{2}\sigma^{-2}\left[\mathbf{y}^\mathsf{T}\mathbf{y}-\mathbf{y}^\mathsf{T}\mathbf{\Phi}\boldsymbol{\mu}\right] \\
&=-\frac{1}{2}\left[\sigma^{-2}\|\mathbf{y}-\mathbf{\Phi}\boldsymbol{\mu}\|^2+\sigma^{-2}\mathbf{y}^\mathsf{T}\mathbf{\Phi}\boldsymbol{\mu}-\sigma^{-2}\boldsymbol{\mu}^\mathsf{T}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi}\boldsymbol{\mu}\right] \\
&=-\frac{1}{2}\left[\sigma^{-2}\|\mathbf{y}-\mathbf{\Phi}\boldsymbol{\mu}\|^2+\boldsymbol{\mu}^\mathsf{T}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}-\sigma^{-2}\boldsymbol{\mu}^\mathsf{T}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi}\boldsymbol{\mu}\right] \\
&=-\frac{1}{2}\left[\sigma^{-2}\|\mathbf{y}-\mathbf{\Phi}\boldsymbol{\mu}\|^2+\boldsymbol{\mu}^\mathsf{T}\mathbf{A}\boldsymbol{\mu}\right]
\end{aligned}
\tag{43}
$$

where $\boldsymbol{\mu}=\sigma^{-2}\mathbf{\Sigma}\mathbf{\Phi}^\mathsf{T}\mathbf{y}$.

Combining Eq. 42 and Eq. 43, the log-likelihood is

$$
\mathcal{L}(\mathbf{y}|\boldsymbol{\alpha},\sigma^2)=\frac{1}{2}\left(\sum_{i=0}^{N}\ln\alpha_i-N\ln\sigma^2+\ln|\mathbf{\Sigma}|-\sigma^{-2}\|\mathbf{y}-\mathbf{\Phi}\boldsymbol{\mu}\|^2-\boldsymbol{\mu}^\mathsf{T}\mathbf{A}\boldsymbol{\mu}\right)
\tag{44}
$$

Then, we compute the derivative

$$
\frac{\partial\mathcal{L}(\mathbf{y}|\boldsymbol{\alpha},\sigma^2)}{\partial\alpha_i}=\frac{1}{2}\left(\alpha_i{}^{-1}-\Sigma_{ii}-\mu_i^2\right).
\tag{45}
$$

Setting $\frac{\partial \mathcal{L}(\mathbf{y}|\boldsymbol{\alpha},\sigma^2)}{\partial \alpha_i} = 0$, we obtain

$$\alpha_i = \frac{1 - \alpha_i \Sigma_{ii}}{\mu_i^2}, \tag{46}$$

where μ_i is the i-th component of $\boldsymbol{\mu}$ in Eq. 13 and Σ_{ii} is the i-th diagonal element of $\boldsymbol{\Sigma}$ in Eq. 13. Note that we have used the derivatives $\frac{\partial}{\partial \alpha_i} \boldsymbol{\mu}^{\mathsf{T}} \mathbf{A} \boldsymbol{\mu} = \mu_i^2$ and

$$\frac{\partial \ln |\boldsymbol{\Sigma}|}{\partial \alpha_i} = -\frac{\partial \ln |\boldsymbol{\Sigma}^{-1}|}{\partial \alpha_i} = -\mathrm{tr}\left\{\boldsymbol{\Sigma}\frac{\partial \boldsymbol{\Sigma}^{-1}}{\partial \alpha_i}\right\} = -\mathrm{tr}\left\{\boldsymbol{\Sigma}\frac{\partial \mathbf{A}}{\partial \alpha_i}\right\} = -\Sigma_{ii}.$$

Define $\gamma_i = 1 - \alpha_i \Sigma_{ii}$, we obtain the update equation for α_i as follows:

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{\mu_i^2} \tag{47}$$

To find σ^2, we compute the derivative

$$\begin{aligned}
\frac{\partial \mathcal{L}(\mathbf{y}|\boldsymbol{\alpha},\sigma^2)}{\partial \sigma^{-2}} &= \frac{1}{2}\left[N\sigma^2 - \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 - \mathrm{tr}\left\{\boldsymbol{\Sigma}\frac{\partial \boldsymbol{\Sigma}^{-1}}{\partial \sigma^{-2}}\right\}\right] \\
&= \frac{1}{2}\left[N\sigma^2 - \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 - \sigma^2\mathrm{tr}\left\{\sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}\right\}\right] \\
&= \frac{1}{2}\left[N\sigma^2 - \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 - \sigma^2\mathrm{tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}^{-1} - \mathbf{A}\right)\right\}\right] \\
&= \frac{1}{2}\left[N\sigma^2 - \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 - \sigma^2\mathrm{tr}\left\{\mathbf{I} - \boldsymbol{\Sigma}\mathbf{A}\right\}\right] \\
&= \frac{1}{2}\left[N\sigma^2 - \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2 - \sigma^2\sum\nolimits_i \gamma_i\right].
\end{aligned} \tag{48}$$

Setting $\frac{\partial \mathcal{L}(\mathbf{y}|\boldsymbol{\alpha},\sigma^2)}{\partial \sigma^{-2}} = 0$, we obtain

$$(\sigma^2)^{\text{new}} = \frac{\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2}{N - \sum_{i=0}^{N}\gamma_i}. \tag{49}$$

Appendix B: Reducing the computation cost

Because the major computation burden of EKM is the computation of H_s PLDA scores in Eq. 5, it is possible to reduce the burden by minimizing the computation time of PLDA scoring. This can be done by pre-computing the terms of the scoring function during the enrollment time as follows. Given a test i-vector $\mathbf{x}_t$ and a target speaker s with H_s enrollment utterances, EKM requires to compute H_s PLDA scores:

$$S_{\mathrm{LR}}(\mathbf{x}_t, \mathbf{x}_{s,j}) = \mathrm{const} + \mathbf{x}_t^{\mathsf{T}}\mathbf{Q}\mathbf{x}_t + \mathbf{x}_{s,j}^{\mathsf{T}}\mathbf{Q}\mathbf{x}_{s,j} + 2\mathbf{x}_t^{\mathsf{T}}\mathbf{P}\mathbf{x}_{s,j}, \quad j = 1, \ldots, H_s \quad (50)$$

where $\mathbf{P}$ and $\mathbf{Q}$ are defined in Eq. 2. Note that the 3rd term $\{\mathbf{x}_{s,j}^{\mathsf{T}}\mathbf{Q}\mathbf{x}_{s,j}\}_{j=1}^{H_s}$ and $\{\mathbf{P}\mathbf{x}_{s,j}\}_{j=1}^{H_s}$ are independent of $\mathbf{x}_t$ and therefore can be pre-computed during enrollment. As a result, the computational complexity of constructing empirical kernel map for each test trial can be reduced from $O(3H_s(D^2 + D))$ to $O(H_s(D^2 + 2D))$, where D is the dimension of i-vector projected by the LDA+WCCN matrix.

Acknowledgment

This work was in part supported by The Hong Kong Research Grant Council, PolyU Grant No. 152117/14E and PolyU Grant No. 152068/15E.

References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEE Trans. on Audio, Speech, and Language Processing 19 (4) (2011) 788–798.

[2] P. Kenny, Bayesian speaker verification with heavy-tailed priors, in: Proc. of Odyssey: Speaker and Language Recognition Workshop, Brno, Czech Republic, 2010.

[3] D. Garcia-Romero, C. Y. Espy-Wilson, Analysis of i-vector length normalization in speaker recognition systems, in: Proc. of Interspeech 2011, Florence, Italy, 2011, pp. 249–252.

[4] S. Prince, J. Elder, Probabilistic linear discriminant analysis for inferences about identity, in: Proc. of 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[5] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Processing 10 (1–3) (2000) 19–41.

[6] M. Mak, W. Rao, Likelihood-ratio empirical kernels for i-vector based PLDA-SVM scoring, in: Proc. ICASSP 2013, Vancouver, Canada, 2013, pp. 7702–7706.

[7] W. Rao, M. W. Mak, Construction of discriminative kernels from known and unknown non-targets for PLDA-SVM scoring, in: Proc. ICASSP 2014, Florence, Italy, 2014.

[8] B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. Muller, G. Ratsch, A. Smola, Input space versus feature space in kernel-based methods, IEEE Trans. on Neural Networks 10 (5) (1999) 1000–1017.

[9] H. Xiong, M. Swamy, M. Ahmad, Optimizing the kernel in the empirical feature space, IEEE Trans. on Neural Networks 16 (2) (2005) 460–474.

[10] S. X. Zhang, M. W. Mak, Optimized Discriminative Kernel for SVM Scoring and Its Application to Speaker Verification, IEEE Trans. on Neural Networks 22 (2) (2011) 173–185.

[11] D. Sturim, D. A. Reynolds, E. Singer, J. P. Campbell, Speaker indexing in large audio databases using anchor models, in: Proc. ICASSP 2001, Salt Lake City, UT, 2001, pp. 429–432.

[12] M. Collet, D. Charlet, F. Bimbot, Speaker tracking by anchor models using speaker segment cluster information, in: Proc. ICASSP 2006, Toulouse, France, 2006, pp. 1009–1012.

[13] E. Noor, H. Aronowitz, Efficient language identification using anchor models and support vector machines, in: Odyssey 2006, San Juan, 2006, pp. 1–6.

[14] Y. Mami, D. Charlet, Speaker recognition by location in the space of reference speakers, Speech Communication 48 (2) (2006) 127–141.

[15] 2012 NIST Speaker Recognition Evaluation, http://www.nist.gov/itl/iad/mig/sre12.cfm (2012).

[16] M. E. Tipping, Sparse Bayesian learning and the relevance vector machine, Journal of Machine Learning Research 1 (2001) 211–244.

[17] H. Tang, Z. X. Chen, T. S. Huang, Comparison of algorithms for speaker identification under adverse far-field recording conditions with extremely short utterances, in: IEEE International Conference on Networking, Sensing and Control, Sanya, 2008.

[18] W. Rao, M. W. Mak, Relevance vector machines with empirical likelihood-ratio kernels for PLDA speaker verification, in: Proc. Int. Sym. on Chinese Spoken Language Processing (ISCSLP 2014), Singapore, 2014.

[19] W. Rao, M. W. Mak, Boosting the performance of i-vector based speaker verification via utterance partitioning, IEEE Trans. on Audio, Speech and Language Processing 21 (5) (2013) 1012 – 1022.

[20] S. Y. Kung, M. W. Mak, S. H. Lin, Biometric Authentication: A Machine Learning Approach, Prentice Hall, Upper Saddle River, New Jersey, 2005.

[21] V. N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[22] C. M. Bishop, Pattern recognition and machine learning, Springer New York, 2006.

[23] H. Yu, M. Mak, Comparison of voice activity detectors for interview speech in NIST speaker recognition evaluation, in: Proc. of Interspeech 2011, Florence, 2011, pp. 2353–2356.

[24] M. Mak, H. Yu, A study of voice activity detection techniques for NIST speaker recognition evaluations, Computer Speech and Language 28 (1) (2014) 295 – 313.

[25] B. S. Atal, Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, J. Acoust. Soc. Am. 55 (6) (1974) 1304–1312.

[26] J. Pelecanos, S. Sridharan, Feature warping for robust speaker verification, in: Proc. of Odyssey: Speaker and Language Recognition Workshop, Crete, Greece, 2001, pp. 213–218.

[27] D. A. Leeuwen, R. Saeidi, Knowing the non-target speakers: The effect of the i-vector population for PLDA training in speaker recognition, in: Proc. ICASSP 2013, Vancouver, BC, Canada, 2013, pp. 6778–6782.

[28] Freesound, http://www.freesound.org/ (Apr. 2005).

[29] http://dnt.kr.hsnr.de/download.html.

[30] M. W. Mak, W. Rao, Utterance partitioning with acoustic vector resampling for GMM-SVM speaker verification, Speech Communication 53 (1) (2011) 119–130.

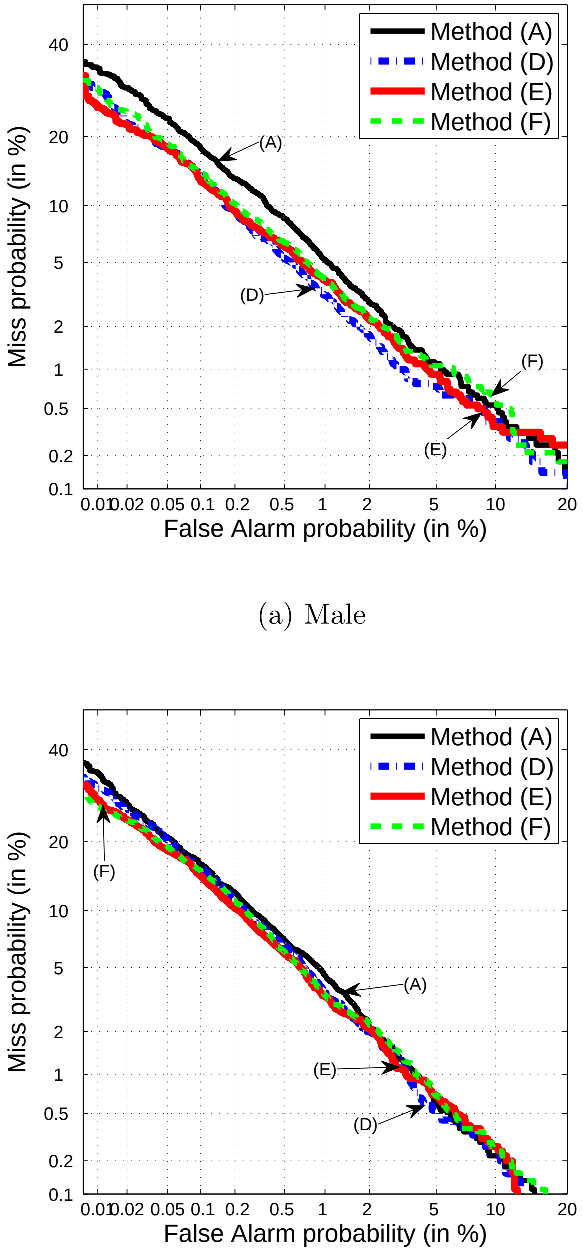
[31] W. Rao, M. W. Mak, Addressing the data-imbalance problem in kernel-based speaker verification via utterance partitioning and speaker comparison, in: Proc. of Interspeech 2011, Florence, 2011, pp. 2717–2720.

[32] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, N. Scheffer, Towards noise-robust speaker recognition using probabilistic linear discriminant analysis, in: Proc. ICASSP 2012, Kyoto, Japan, 2012, pp. 4253–4256.

[33] A. Hatch, S. Kajarekar, A. Stolcke, Within-class covariance normalization for SVM-based speaker recognition, in: Proc. of the 9th International Conference on Spoken Language Processing, Pittsburgh, PA, USA, 2006, pp. 1471–1474.

[34] M. McLaren, M. I. Mandasari, D. A. van Leeuwen, Source normalization for language-independent speaker recognition using i-vectors, in: Proc. of Odyssey: Speaker and Language Recognition Workshop, Singapore, 2012.

[35] *http://www.miketipping.com/sparsebayes.htm.*



(a) Male

(b) Female

Figure 3: The DET performance of different scoring methods in CC2 of NIST 2012 SRE: (A) PLDA, (D) PLDA-SVM, (E) PLDA-RVM classification, and (F) PLDA-RVM regression. See Table 1 for the nomenclature of methods in the legend.

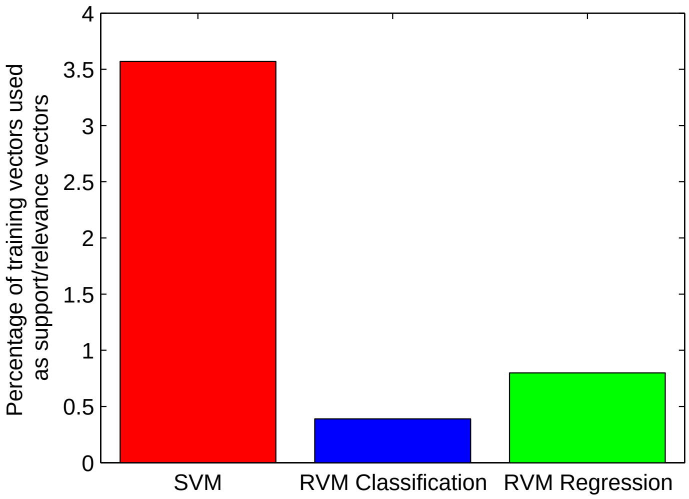


Figure 4: Sparseness of SVM and RVM (percentage of training vectors used as support/relevance vectors in resulting models).