Relevance Vector Machines with Empirical Likelihood-Ratio Kernels 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 works have shown the benefits of empirical likelihood ratio (LR) kernels for i-vector/PLDA speaker verification. The method not only utilizes the multiple enrollment utterances of target speakers effectively, but also opens up opportunity for adopting sparse kernel machines for PLDA-based speaker verification systems. This paper proposes taking the advantages of the empirical LR kernels by incorporating them into relevance vector machines (RVMs). Results on NIST 2012 SRE demonstrate that the performance of RVM regression equipped with empirical LR kernels is slightly better than that of the support vector machines after performing utterance partitioning.

Index Terms: Relevance Vector Machines, Empirical LR kernel, Probabilistic Linear Discriminant Analysis, I-vectors, NIST SRE.

1. Introduction

Nowadays, utilizing i-vectors [1] as features and probabilistic linear discriminant analysis (PLDA) [2–4] 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] and the total variability matrix. The implicit use of background information is a drawback of this method.

To address the limitation of PLDA scoring, an empirical LR kernel that takes the background speaker information *explicitly* during the scoring process was proposed in [6, 7]. This method captures the discrimination between a target-speaker and background-speakers in the SVM weights as well as in the score vectors that live in an empirical score space. Specifically, for each target speaker, an empirical score space with dimension equal to the number of training i-vectors for this target speaker is defined by using the idea of empirical kernel maps [8–10]. 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 training i-vectors. A speaker-dependent SVM – referred to as empirical LR 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.

NIST 2012 SRE [11] introduces some new protocols that help researchers to enhance system performance. One of the new protocols is that some target speakers have multiple enrollment utterances. Common approaches to deal with multiple enrollment utterances include averaging the i-vectors of utterances and averaging the PLDA scores of these i-vectors. Both methods achieve similar performance. But recent research [12] suggests that the former is slightly better. Adopting empirical LR kernels in the SVM scoring is another way to use the multiple enrollment i-vectors. When the SVM kernel ($\mathbb{K}$ in Eq. 3) is linear, this scoring method is equivalent to computing the weighted average of the PLDA scores. Thus, SVM scoring with empirical LR kernel can be considered as a generalization of score averaging.

Therefore, adopting empirical LR kernels in SVM scoring not only utilizes the multiple enrollment utterances of target speakers, but also opens up opportunity for adopting sparse kernel machines in PLDA-based speaker verification systems. Accordingly, this paper proposes incorporate the empirical LR kernel into a sparse kernel machine known as the relevance vector machine (RVM) [13]. 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 [13], but can obtain a comparable performance as SVM. Experiments on NIST 2012 SRE demonstrate that the performance of RVM regression with empirical LR kernels after using utterance partitioning (UP-AVR) to generate more training i-vectors [14–16] is slightly better than that of SVM.

RVMs have been applied to speaker identification. For example, [17] compares 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.

2. Empirical Likelihood-Ratio Kernels

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

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

where

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

This work was in part supported by The Hong Kong Polytechnic University 5-ZH90 and G-YN18.

978-1-4799-4219-0/14/$31.00 ©2014 IEEE

and $\mathbf{V}$ is the factor loading matrix and $\boldsymbol{\Sigma}$ 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 $\mathbf{V}$ and $\boldsymbol{\Sigma}$. To make better use of the background information, an empirical likelihood-ratio (LR) kernel based on the idea of empirical kernel map [8–10] was derived in [6, 7].

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 [16] to produce multiple i-vectors from his/her enrollment utterance. Denote these i-vectors as $\mathcal{X}_s = \{\mathbf{x}_{s,1}, \ldots, \mathbf{x}_{s,H_s}\}$ and the set of background-speaker i-vectors as $\mathcal{X}_b = \{\mathbf{x}_{b,1}, \ldots, \mathbf{x}_{b,B}\}$. Therefore, $\mathcal{X} = \{\mathcal{X}_s, \mathcal{X}_b\}$ are the training set for target-speaker s. The empirical likelihood-ratio kernel is given by:

$$K(\mathbf{x}_t, \mathbf{x}_{s,j}) = \mathbb{K}\left(\overrightarrow{S}_{\text{LR}}(\mathbf{x}_t, \mathcal{X}), \overrightarrow{S}_{\text{LR}}(\mathbf{x}_{s,j}, \mathcal{X})\right) \quad (3)$$

where

$$\overrightarrow{S}_{\text{LR}}(\mathbf{x}_t, \mathcal{X}) = \begin{bmatrix} S_{\text{LR}}(\mathbf{x}_t, \mathbf{x}_{s,1}) \\ \vdots \\ S_{\text{LR}}(\mathbf{x}_t, \mathbf{x}_{s,H_s}) \\ S_{\text{LR}}(\mathbf{x}_t, \mathbf{x}_{b,1}) \\ \vdots \\ S_{\text{LR}}(\mathbf{x}_t, \mathbf{x}_{b,B'}) \end{bmatrix} \quad (4)$$

is an empirical kernel map, $S_{\text{LR}}(\mathbf{x}_t, \mathbf{x}_{s,i})$ is a PLDA score and $\mathbb{K}(\cdot, \cdot)$ is a standard kernel function, e.g., linear or RBF. $\overrightarrow{S}_{\text{LR}}(\mathbf{x}_{s,j}, \mathcal{X})$ can be obtained by replacing $\mathbf{x}_t$ in Eq. 4 with $\mathbf{x}_{s,j}$. B' $(\leq B)$ is the number of background i-vectors selected from the background speaker set $\mathcal{X}_b$.

The score vector in Eq. 4 contains the LR scores of $\mathbf{x}_t$ with respect to the background i-vectors. As a result, discriminative information between same-speaker pairs $\{\mathbf{x}_t, \mathbf{x}_{s,j}\}_{j=1}^{H_s}$ and different-speaker pairs $\{\mathbf{x}_t, \mathbf{x}_{b,j}\}_{j=1}^{B'}$ is embedded in the score vector. Note that the vector size in Eq. 4 is independent of the number of target-speakers. Therefore, the method is scalable to large systems with thousands of speakers.

3. SVM with Empirical LR Kernels

Support vector machine is a well-known supervised learning method used for classification and regression. Assume that we are given N training vectors $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ with labels $y_n \in \{+1, -1\}, n = 1, ..., N$. Using the pairs $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$, an SVM can be trained [18]. Given a test vector $\mathbf{x}_t$, the SVM's output is written as

$$f(\mathbf{x}_t; \mathbf{w}) = \sum_{i=1}^{N} w_i K(\mathbf{x}_t, \mathbf{x}_i) + w_0 \quad (5)$$

where $\mathbf{w} = [w_0, ..., w_N]$ 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 applies the empirical LR kernel (Eq. 3) to SVMs. Specifically, Eq. 5 is rewritten as

$$\begin{aligned} S_{\text{SVM}}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = &\sum_{i \in \text{SV}_s} \alpha_{s,i} K(\mathbf{x}_t, \mathbf{x}_{s,i}) - \\ &\sum_{j \in \text{SV}_b} \alpha_{b,j} K(\mathbf{x}_t, \mathbf{x}_{b,j}) + w_0 \end{aligned} \quad (6)$$

where SV_s and SV_b contain the indexes of the support vectors corresponding to the speaker class and impostor class, respectively. $\alpha_{s,i}$ and $\alpha_{b,j}$ are the Lagrange multipliers of the SVM.

While our earlier studies [6, 7] have demonstrated that the empirical LR kernel works very well for the SVMs and that PLDA-SVM scoring (Eq. 6) performs better than simple PLDA-LR scoring (Eq. 1), the SVMs in Eq. 6 still has some limitations [13]. First, 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 that we can use for training the SVMs. Second, the SVM scores in Eq. 6 are not probabilistic, meaning that score normalization may be need to adjust the score range of individual SVMs. Third, to achieve the best performance, it is necessary to tradeoff the training error and the margin of separation through adjusting 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. As a result, [6, 7] used the same penalty factor for all target speakers. In our experiments, the penalty factor C was set as 1.

4. RVM with Empirical LR Kernels

In terms of output scoring, relevance vector machines [13] and support vector machines have the same form (Eq. 5). However, their learning methods are very different. The training of SVMs is based on structural risk minimization, whereas RVM training is based on Bayesian relevance learning [13]. The main goal of RVMs is to overcome some of limitations in SVMs mentioned above in Section 3.

RVM is a Bayesian treatment of Eq. 5. When an RVM is applied to regression, the target y is assumed to follow a Gaussian distribution with mean $f(\mathbf{x}; \mathbf{w})$ and variance σ^2; when it is applied to classification, the target conditional distribution $p(y|\mathbf{x})$ is assumed to follow a Bernoulli distribution. This paper focuses on the regression mode of RVMs because it performs significantly better than the classification mode in NIST SRE (see Section 5).

Assume that for target speaker s, we have a set of training i-vectors $\mathcal{X} = \{\mathcal{X}_s, \mathcal{X}_b\}$ as in Section 2 and that $y_i = 1$ when $\mathbf{x}_i \in \mathcal{X}_s$ and $y_i = -1$ when $\mathbf{x}_i \in \mathcal{X}_b$. Assume also that y_i's $(i = 1, ..., N)$ are independent, the likelihood of the training data set can be written as [13]:

$$p(\mathbf{y}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|^2\right\} \quad (7)$$

where

$$\begin{aligned} &N = |\mathcal{X}_s| + |\mathcal{X}_b|; \ \mathbf{y} = [y_1, ..., y_N]^\mathsf{T} \\ &\mathbf{w} = [w_0, ..., w_N]^\mathsf{T}; \ \boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), ..., \boldsymbol{\phi}(\mathbf{x}_N)]^\mathsf{T} \\ &\boldsymbol{\phi}(\mathbf{x}_i) = [1, K(\mathbf{x}_i, \mathbf{x}_1), K(\mathbf{x}_i, \mathbf{x}_2), ..., K(\mathbf{x}_i, \mathbf{x}_N)]^\mathsf{T} \end{aligned} \quad (8)$$

and σ^2 is the variance of the additive Gaussian noise ϵ in the model $y = f(\mathbf{x}; \mathbf{w}) + \epsilon$. To avoid over-fitting, RVM defines a zero-mean Gaussian prior distribution over $\mathbf{w}$:

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=0}^{N} \mathcal{N}(w_i|0, \alpha_i^{-1}) \quad (9)$$

where $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, ..., \alpha_N]^\mathsf{T}$ and α_i is the hyperparameter associated with weight w_i. By considering $\mathbf{w}$ probabilistic and using the notion of conditional independence [19], the predictive distribution of y_t given a test vector $\mathbf{x}_t$ is

$$p(y_t|\mathbf{y}) = \int p(y_t|\mathbf{w}, \boldsymbol{\alpha}, \sigma^2)p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{y})d\mathbf{w}d\boldsymbol{\alpha}d\sigma^2 \quad (10)$$

where

$$p(y_t|\mathbf{w}, \boldsymbol{\alpha}, \sigma^2) = p(y_t|\mathbf{w}, \sigma^2) \quad (11)$$

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{y}) = p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\mathbf{y}) \quad (12)$$

After some derivations [13, 19], we can obtain the posterior distribution over the weights as follows:

$$p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha}, \sigma^2) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (13)$$

where

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathsf{T}}\mathbf{y} \\ \boldsymbol{\Sigma} &= (\sigma^{-2}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}+\mathbf{A})^{-1};\ \mathbf{A} = \mathrm{diag}(\alpha_0, \alpha_1, ..., \alpha_N). \end{aligned} \quad (14)$$

Instead of computing the posterior $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{y})$ in Eq. 11, [13] uses a delta function at the most probable values of $\boldsymbol{\alpha}$ and σ^2 as an approximation. Therefore, using Eq. 12 and assuming uniform priors for $\boldsymbol{\alpha}$ and σ^2, Eq. 10 reduces to

$$p(y_t|\mathbf{y}) = \int p(y_t|\mathbf{w}, \boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}})p(\mathbf{w}|\mathbf{y}, \boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}})d\mathbf{w} \quad (15)$$

where

$$\begin{aligned} (\boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}}) &= \arg\max_{\boldsymbol{\alpha}, \sigma^2} p(\boldsymbol{\alpha}, \sigma^2|\mathbf{y}) \\ &= \arg\max_{\boldsymbol{\alpha}, \sigma^2} \int p(\mathbf{y}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w}. \end{aligned} \quad (16)$$

Because both terms in the integrand of Eq. 15 are Gaussian, the predictive distribution is also a Gaussian:

$$p(y_t|\mathbf{y}, \boldsymbol{\alpha}_{\text{MP}}, \sigma^2_{\text{MP}}) = \mathcal{N}(y_t|g(\mathbf{x}_t), \sigma^2_t) \quad (17)$$

with

$$g(\mathbf{x}_t) = \boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{x}_t) \quad (18)$$

$$\sigma^2_t = \sigma^2_{\text{MP}} + \boldsymbol{\phi}(\mathbf{x}_t)^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{\phi}(\mathbf{x}_t) \quad (19)$$

The two optimized hyperparameters $\boldsymbol{\alpha}_{\text{MP}}$ and σ^2_{MP} can be obtained by maximum likelihood. Readers may refer to Section 2.3 in [13] for the details of the optimization. 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**.

In our case, we used $g(\mathbf{x}_t)$ in Eq. 18 as the output of RVM regression. After incorporating the empirical LR kernel (Eq. 3), the score of RVM regression can be written as

$$S_{\text{RVM}}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = g(\mathbf{x}_t) = \boldsymbol{\mu}^T\boldsymbol{\phi}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) \quad (20)$$

where

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{x}_t, \mathcal{X}_s, \mathcal{X}_b) = [&1, K(\mathbf{x}_t, \mathbf{x}_{s,1}), ..., K(\mathbf{x}_t, \mathbf{x}_{s,H_s}), \\ &K(\mathbf{x}_t, \mathbf{x}_{b,1}), ..., K(\mathbf{x}_t, \mathbf{x}_{b,B})]^{\mathsf{T}} \end{aligned} \quad (21)$$

5. Experiments and Results

5.1. Speech Data and PLDA Models

The male *core set* (common evaluation condition 2) of NIST 2012 Speaker Recognition Evaluation (SRE) [11] was used for performance evaluation. We removed the 10-second and the summed-channel utterances from the training segments of the SRE but ensured that all target speakers have at least one long utterance for training. The speech files of male speakers in NIST 2005–2010 SREs were used as development data for training the UBM, total variability matrix, LDA-WCCN, PLDA models. The silence regions were removed by a VAD [20, 21]. 19 MFCCs together with energy plus their 1st- and 2nd- derivatives were extracted from the speech regions, followed by cepstral mean normalization [22] and feature warping [23] with a window size of 3 seconds. A 60-dim acoustic vector was extracted every 10ms, using a Hamming window of 25ms.

The i-vector systems are based on a gender-dependent UBM with 1024 mixtures. We applied whitening [24] and i-vector length normalization [3] to the 400-dimensional i-vectors. Then, we performed linear discriminant analysis (LDA) [19] and within-class covariance normalization (WCCN) [24] on the resulting vectors to reduce the dimension to 200 before training the PLDA models with 150 latent variables.

5.2. Property of Empirical LR Kernels in SVM and RVM

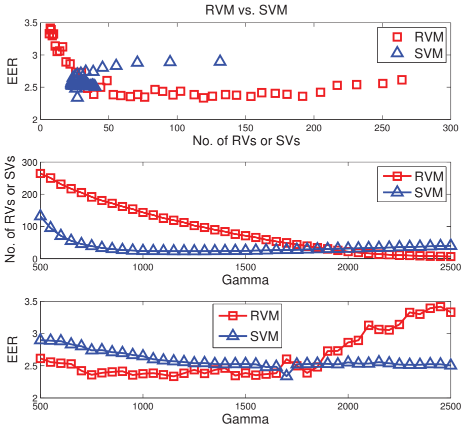

Figure 1: The property of empirical LR kernels in SVMs and RVM regressions. Gamma is the RBF parameter γ.

We extracted 108 target speakers with true-target trials and imposter trials from 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 $\mathbb{K}(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\gamma^2})$ was adopted, where the RBF parameters γ was varied from 500 to 2500.[1]

The top panel of Fig. 1 plots the average EER against the average number of SVs and RVs in the SVMs and RVMs. It clearly shows that when the number of SVs increases, the performance of SVMs becomes poor. On the other hand, while the performance of RVMs is poor when the number of RVs is very small, their performance is fairly stable and is better than that of the SVMs once the number of relevance vectors is sufficient.

The middle panel of Fig. 1 shows that when the RBF parameter γ increases, the number of SVs decreases first and then gradually increases. On the other hand, the number of RVs monotonically decreases when γ increases. More importantly, for

[1] Because the LR scores have range between -579.5 to 199.8 and the dimension of the LDA-projected i-vector is 150, a large value of γ is necessary.

a wide range of γ, there are more RVs than SVs, suggesting that for this dataset, the RVMs (under the regression mode) is less sparse than the SVMs. This phenomenon is attributed to the fact that the structural risk minimization attempts to find a small number of SVs that lie on the margin or the wrong side of it, whereas the Bayesian relevance learning attempts to find a set of weights that maximize the likelihood in Eq. 7.

The middle and bottom panels of Fig. 1 suggest that there is a lower limit on the number of RVs for the RVMs to be effective. In our experiments, this limit is around 50. Below this value, the performance of RVMs deteriorates rapidly and becomes significantly inferior to the SVMs. However, once the RVMs have sufficient RVs, their performance can be better than that of the SVMs.

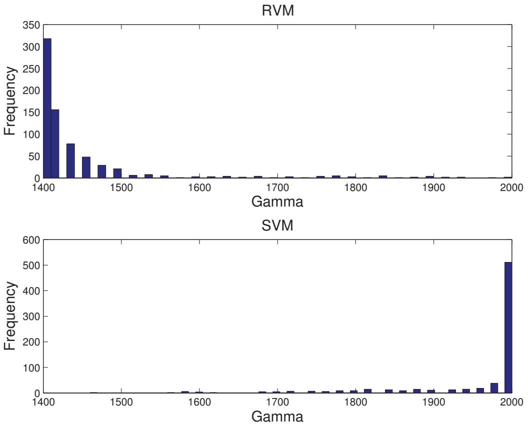

Figure 2: Histograms of the RBF parameter γ in target-speakers' RVMs (top panel) and SVMs (bottom panel).

5.3. Optimization of RBF Parameter Gamma (γ)

Because the performance of RVMs depends on RBF parameter γ, an optimization procedure was developed to find an appropriate γ for each target-speaker's RVM. Specifically, for each target speaker, a development set was created by applying utterance partitioning [16] on his/her enrollment utterances to generate a number of enrollment i-vectors. Then, some of these i-vectors were used for training an RVM. The remaining i-vectors were considered as true-target trials and the utterances of background speakers were considered as impostor trials. During RVM training, the value of γ was varied from 1400 to 2000 and the true-speaker scores and impostor scores were computed. The procedure continues until the difference between the mean of the true-target scores and the mean of the impostor scores is maximum.

Fig. 2 shows the histograms of RBF parameter γ in the target-speakers' RVMs and SVMs. It shows that the preferred value of γ for RVMs is between 1400 and 1500 and that for SVMs is between 1900 and 2000. These ranges of values also agree with those in Fig. 1.

5.4. Performance Comparison

Table 1 shows that the performance of RVM classification with empirical LR kernel is poor and even worse than that of the baseline (Gaussian PLDA). The poor performance is caused by the severe sparsity of the RVM classification models. They are so sparse that the average number of relevance vectors per RVM is only two. In other words, each class only contains one relevance vector. Furthermore, RVM classification applies a logistic link function to compute the probabilistic outputs (posterior probabilities of the target-speaker class). 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. 5 was used for computing the verification scores in the classification mode of RVMs and SVMs in our experiments.

Table 1 also shows that adopting empirical LR kernels in both SVM classification and RVM regression can improve performance. In addition, without performing the utterance partitioning (UP-AVR), the performance of RVM regression is comparable with SVM. However, after performing the UP-AVR, the performance of both RVM regression and SVM improves and RVM regression slightly outperforms SVM. This results also agrees with the conclusion in Fig. 1 that the performance of RVM regression will be better than the performance of SVM once the number of relevance vectors is sufficient.

Method	EER (%)	MinNDCF
PLDA	2.40	0.33
PLDA+UP-AVR	2.32	0.32
PLDA+SVM	2.07	0.31
PLDA+RVM-C	3.76	0.48
PLDA+RVM-R	2.32	0.28
PLDA+UP-AVR+SVM	1.97	0.30
PLDA+UP-AVR+RVM-C	3.00	0.42
PLDA+UP-AVR+RVM-R	**1.94**	**0.28**

Table 1: Performance comparison between SVM and RVM in common condition 2 of NIST 2012 SRE. *RVM-C* represents relevance vector machine classification. *RVM-R* represents relevance vector machine regression. *UP-AVR* represents utterance partitioning with acoustic vector resampling [16]. 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 training SVMs that use empirical LR kernel (Eq. 3 and Eq. 4).

6. Conclusions

This paper investigates the property of empirical LR kernels in SVM and RVM and compares the performance between these two sparse kernel machines in PLDA-based speaker verification. Experimental results show that RVM classification is not appropriate for the verification task in NIST SRE, but RVM regression combining with empirical LR kernel can achieve comparable performance as SVM. In addition, this paper also suggests that UP-AVR can boost the performance of both SVM and RVM regression and the performance of RVM regression is slightly better than SVM after adopting UP-AVR. 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.

7. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

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

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

[4] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. of 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, Oct. 2007.

[5] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, Jan. 2000.

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

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

[8] B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. R. Muller, G. Ratsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Trans. on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, Sept. 1999.

[9] H. Xiong, M. Swamy, and M. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Trans. on Neural Networks*, vol. 16, no. 2, pp. 460 – 474, 2005.

[10] S. X. Zhang and M. W. Mak, "Optimized Discriminative Kernel for SVM Scoring and Its Application to Speaker Verification," *IEEE Trans. on Neural Networks*, vol. 22, no. 2, pp. 173–185, 2011.

[11] "*http://www.nist.gov/itl/iad/mig/sre12.cfm*."

[12] P. Rajan, T. Kinnunen, and V. Hautamaki, "Effect of multicondition training on i-vector PLDA configurations for speaker recognition," in *Proc. of Interspeech 2013*, Lyon, France, Aug. 2013.

[13] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[14] W. Rao and 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, Aug. 2011, pp. 2717–2720.

[15] M. W. Mak and W. Rao, "Utterance partitioning with acoustic vector resampling for GMM-SVM speaker verification," *Speech Communication*, vol. 53, no. 1, pp. 119–130, Jan. 2011.

[16] W. Rao and M. W. Mak, "Boosting the performance of i-vector based speaker verification via utterance partitioning," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 21, no. 5, pp. 1012 – 1022, 2013.

[17] H. Tang, Z. X. Chen, and 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, Apr. 2008.

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

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

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

[21] M. Mak and H. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer Speech and Language*, vol. 28, no. 1, pp. 295 – 313, Jan. 2014.

[22] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Am.*, vol. 55, no. 6, pp. 1304–1312, Jun. 1974.

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

[24] A. Hatch, S. Kajarekar, and 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, Sep. 2006, pp. 1471–1474.