Optimized Discriminative Kernel for SVM Scoring and its Application to Speaker Verification

Shi-Xiong Zhang, *Student Member* and Man-Wai Mak, *Member*

***Abstract*—The decision making process of many binary classification systems is based on the likelihood-ratio (LR) scores of test patterns. This paper shows that LR scores can be expressed in terms of the similarity between the supervectors formed by stacking the mean vectors of Gaussian mixture models corresponding to the test patterns, the target model, and the background model. By interpreting the SVM kernels as a specific similarity (or discriminant) function between supervectors, this paper shows that LR scoring is a special case of SVM scoring and that most sequence kernels can be obtained by assuming a specific form for the similarity function of supervectors. The paper further shows that this assumption can be relaxed to derive a new general kernel. The kernel function is general in that it is a linear combination of any kernels belonging to the reproducing kernel Hilbert space. The combination weights are obtained by optimizing the ability of a discriminant function to separate the positive- and negative-classes using either regression analysis or SVM training. The idea was applied to both high- and low-level speaker verification. In both cases, results show that the proposed kernels achieve a better performance than several state-of-the-art sequence kernels. Further performance enhancement was also observed when the high-level scores were combined with acoustic scores.**

***Index Terms*—Support vector machines; kernel optimization; sequence kernels; speaker verification.**

I. Introduction

Speaker verification is a binary classification problem in which a person's identity is verified based on his/her voice. Current implementations of speaker verification typically use Gaussian mixture models (GMM) [2] to represent the low-level acoustic characteristics of speakers via extracting the frame-based mel-frequency cepstral coefficients (MFCCs) [3] from their speech. One drawback of using low-level features is that they are sensitive to background noise and channel effects. Over the years, various approaches to overcoming this drawback have been proposed. These approaches can be divided into feature transformation [4], [5], model transformation [6], score normalization [7], model-based projection [8], [9], factor analysis [10], and long-term, high-level features modeling [11]–[14].

In most of these speaker verification systems (e.g. acoustic-based GMM-UBM [2] and articulatory feature-based n-gram (AFCPM) [14]), scoring is done by computing the log-likelihood ratio given the speech signal of a claimant. More specifically, each frame of speech is scored independently against a speaker-dependent generative model and a universal background model (UBM), and the resulting frame-based likelihood ratio (LR) scores are accumulated to produce an utterance-based score for decision making. A drawback of this approach is that the target-speaker model and the UBM are trained *separately* to maximize the likelihood of the speaker-class data and impostor-class data. To mitigate this drawback, a number of sequence kernels—such as the generalized linear discriminant sequence (GLDS) kernel [15], n-gram kernel [16], linearized LR kernel [17], GMM-supervector (GSV) kernel [18], and Fisher kernel [19]—have been proposed for speaker verification. All of these kernels can convert variable length sequences into fixed-dimension vectors for classification (or scoring) by support vector machines (SVM) [20]. They are derived from similarity metrics between two sequences by assuming a specific form for the similarity (or discriminant) functions. A key advantage of these kernel-based approach is that the discriminative information of the speaker- and impostor-class data can be harnessed via the speaker-dependent SVMs.

In this paper, we first argue that likelihood-ratio scoring is a special case of SVM scoring. We use this relationship to explain why SVM-based speaker verification systems usually perform better than conventional GMM-UBM systems. Then, we further generalize the SVM scoring by relaxing the form of the similarity function used by the SVMs. More specifically, instead of assuming a fixed form for the discriminant or scoring functions, we use a linear combination of kernel functions in the reproducing kernel Hilbert space as the discriminant function. We show that the optimal combination weights of the discriminant function can be obtained by solving a functional optimization problem using regression analysis, leading to a kernel that is a general form of the GLDS, GSV, linearized LR or n-gram kernel. We further demonstrate that the combination weights can also be optimized by the SVM training algorithm. Then, using the idea of empirical kernel map [21]–[23], the optimized discriminant function can satisfy the Mercer's condition [24] and be used as a kernel for SVM scoring.

The main contribution of this paper is as follows. Unlike the conventional GMM-SVM approach where only the Lagrange multipliers of the scoring SVM are optimized, our method also optimizes the combination weights that constitute the kernel. This idea of double optimization is applied to both low- and high-level speaker verification. For the former, the discriminant function is a linear combination of the GMM-supervector kernels. For the latter, the discriminant function is a linear combination of linearized LR kernels [17]. In

S. X. Zhang and M. W. Mak are with Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University. This work was in part supported by Center for Multimedia Signal Processing, The Hong Polytechnic University (4-ZZ7W) and Research Grant Council of the Hong Kong SAR (PolyU 5251/08E). This paper is an enhanced version of the authors' preliminary work presented in Interspeech 2009 [1].

both cases, the combination weights are determined by either regression analysis or soft-margin maximization. Evaluations show that the proposed kernel scoring approach is superior to conventional SVM scoring and LR scoring. In particular, the double optimization procedure can effective explore the discriminative information among speakers, resulting in more discriminative SVMs than those based on single optimization. It was also found that the kernel-based SVM scoring and likelihood ratio scoring are complementary to each other, leading to better performance when they are combined. Further performance enhancement was also observed when the high-level feature-based scores were combined with acoustic scores.

Notational Convention. Throughout the paper, boldface lowercase letters represent vectors and boldface uppercase letters represent matrices. Italic letters with an arrow on top represent supervectors. Subscribes s, c, and b represent target speakers, claimants, and background speakers, respectively. For example, $\overrightarrow{A_s}$, $\overrightarrow{A_b}$, and $\overrightarrow{A_{b_k}}$ denote the supervector of speaker s, the universal background model, and the k-th background speaker, respectively.

II. Supervector-Based Framework for Speaker Verification

A. Supervector-Based Likelihood Ratio Scoring

In speaker verification, speech utterances are typically represented by variable-length observations $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$. Given the observations $\boldsymbol{O}_c$ of claimant c , many speaker verification systems compute the utterance-based score $S_{\text{LR}}(\boldsymbol{O}_c, s)$ of claimant c for target speaker s by accumulating the frame-based likelihood ratio (LR) scores. More specifically, each frame of speech is scored independently against a speaker-dependent generative model $p_s(\boldsymbol{x})$ and a universal background model (UBM) $p_b(\boldsymbol{x})$, and the resulting frame-based LR scores are accumulated to produce an utterance-based score for decision making:

$$S_{\text{LR}}(\boldsymbol{O}_c, s) = \frac{1}{T}\sum_{t=1}^{T} \log \frac{p_s(\boldsymbol{o}_t)}{p_b(\boldsymbol{o}_t)}, \text{ where } \boldsymbol{o}_t \in \boldsymbol{O}_c. \quad (1)$$

It can be shown that for both discrete (e.g. n-gram models) and continuous (e.g. GMMs) cases, the LR scores can be expressed via the similarity between the supervectors obtained from the models of target speaker s, background speakers and the test utterance of claimant c:[1]

$$\begin{aligned} S_{\text{LR}}(\boldsymbol{O}_c, s) &= \frac{1}{T}\sum_{t=1}^{T} \log \frac{p_s(\boldsymbol{o}_t)}{p_b(\boldsymbol{o}_t)} \\ &\doteq f\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) - f\left(\overrightarrow{A_c}, \overrightarrow{A_b}\right) + d_s, \end{aligned} \quad (2)$$

where $\overrightarrow{A}$ is a supervector formed by stacking the parameters of the corresponding generative model, $f(\cdot,\cdot)$ is a similarity function and d_s is a bias. The definition of $\overrightarrow{A}$, f and d_s for different models are summarized in Table I. See the appendix for a derivation of Eq. 2 for continuous generative models and refer to [17] for discrete models. Fig. 1 illustrates the supervector-based implementation of LR scoring.

Obviously, the supervectors derived from continuous generative models are different from those derived from discrete models. Specifically, for GMM-UBM, given a test utterance from claimant c, a GMM is created by adapting the UBM using maximum a posteriori (MAP) adaptation [2]; a supervector $\overrightarrow{A_c}$ is then constructed by stacking the mean vectors of the GMM. The supervector $\overrightarrow{A_s}$ for speaker s is constructed in a similar manner. The supervector $\overrightarrow{A_b}$ is constructed by stacking the mean vectors of the UBM. For n-gram models, the supervectors are constructed by stacking the probabilities of different n-gram combinations.

Note that because the LR function comes from a Bayesian framework, Eq. 2 is only valid for probability-based (generative) models.

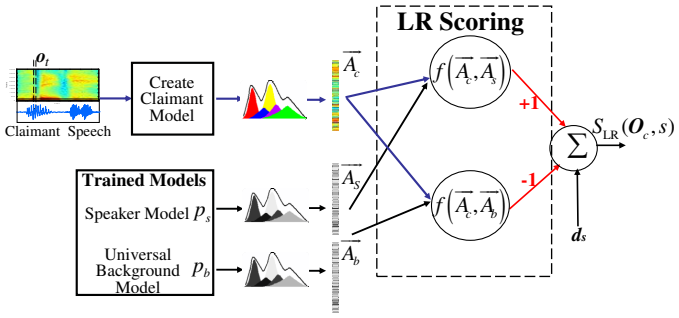

Fig. 1. The supervector-based implementation of LR scoring in speaker verification. The LR score can be obtained by computing the similarity between claimant's supervector and target speaker's supervector minus the similarity between claimant's supervector and background supervector, where the similarity (or discriminant) function f and bias d_s are defined in Table I.

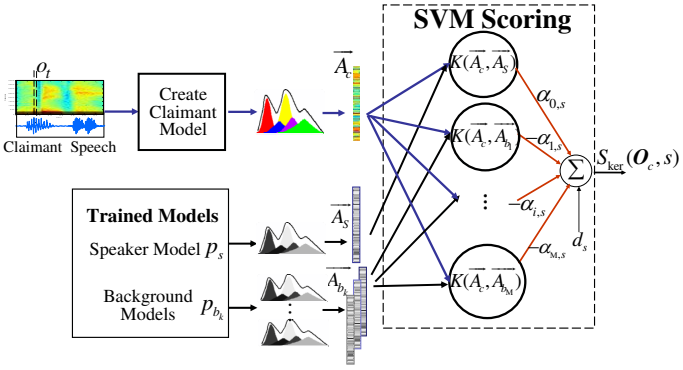

Fig. 2. SVM scoring in speaker verification. $K(\cdot,\cdot)$ could be any of the six kernels discussed in Sections III and IV. See Fig. 1 for a comparison between kernel-based SVM scoring and LR scoring.

B. Supervector-based SVM Scoring

Fig. 1 and Eq. 2 suggest three possible improvements of LR scoring:

1) Replacing the fixed multiplication factors '+1' and '−1' by weights that are optimally determined by SVM training.

[1]For continuous models, Eq. 2 holds under the conditions that the number of training vectors is significantly larger than the relevance factor in MAP adaptation and that only one iteration of MAP adaptation is performed (see the appendix for details).

TABLE I

DEFINITION OF $\overrightarrow{A}$ AND f FOR DIFFERENT TYPES OF MODELS IN COMPUTING $S_{\text{LR}}(\boldsymbol{O}_c, s)$ IN EQ. 2. $\boldsymbol{\mu}_i$ IS THE MEAN OF THE i-TH GAUSSIAN; $\boldsymbol{Q}_{\text{GSV}} = diag\left\{\lambda_{b,1}^{-1} diag\left(\boldsymbol{\Sigma}_{b,1}\right), \ldots, \lambda_{b,G}^{-1} diag\left(\boldsymbol{\Sigma}_{b,G}\right)\right\}$, WHERE $\lambda_{b,i}$ AND $\boldsymbol{\Sigma}_{b,i}$ ARE THE MIXTURE WEIGHT AND COVARIANCE MATRIX OF THE i-TH GAUSSIAN IN THE UBM, AND G IS THE NUMBER OF GAUSSIANS; $f_{s,s} \triangleq f\left(\overrightarrow{A_s}, \overrightarrow{A_s}\right)$; $\Pr(i)$ IS THE PROBABILITY OF OCCURRENCES OF THE i-TH COMBINATIONS IN N-GRAMS; R IS THE NUMBER OF COMBINATIONS; $\ell_{\boldsymbol{o}_t}$ IS THE PHONE LABEL OF $\boldsymbol{o}_t$; $\log \overrightarrow{A_s}/\overrightarrow{A_b}$ MEANS ELEMENT-WISE DIVISION FOLLOWED BY LOGARITHM.

Supervector Formulation of LR: $S_{\text{LR}}(\boldsymbol{O}_c, s) = \frac{1}{T}\sum_{t=1}^{T} \log p_s(\boldsymbol{o}_t)/p_b(\boldsymbol{o}_t) \doteq f\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) - f\left(\overrightarrow{A_c}, \overrightarrow{A_b}\right) + d_s$				
Type of Generative Models	PDF $p(\boldsymbol{o}_t)$	Supervector $\overrightarrow{A}$	Similarity $f\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right)$	Bias d_s
GMM-UBM [2]	$\sum_{i=1}^{G} \lambda_i \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$	$[\boldsymbol{\mu}_1^{\text{T}}, \ldots, \boldsymbol{\mu}_G^{\text{T}}]^{\text{T}}$	$\overrightarrow{A_c}^{\text{T}} \boldsymbol{Q}_{\text{GSV}}^{-1} \overrightarrow{A_s}$	$-\frac{f_{s,s} - f_{b,b}}{2}$
n-gram Model [14], [25]	$\Pr(i = \ell_{\boldsymbol{o}_t})$	$[\Pr(1), \ldots, \Pr(R)]^{\text{T}}$	$\left\langle \overrightarrow{A_c}, \log \overrightarrow{A_s}/\overrightarrow{A_b} \right\rangle$	0

2) Replacing the function f that measures the similarity between input supervectors by a suitable kernel, e.g., the GSV kernel [18] or GLDS kernel [15].
3) The LR scoring in Fig. 1 contains two processing nodes. More discriminative information may be obtained by adding extra processing nodes that evaluates the difference between the claimant's speech ($\overrightarrow{A_c}$) and each of the background speakers $\overrightarrow{A_{b_i}}$.

These improvements lead to the SVM scoring shown in Fig. 2. The SVM output in Fig. 2 can be considered as a scoring function:

$$S_{\text{ker}}(\boldsymbol{O}_c, s) = \alpha_{0,s} K(\overrightarrow{A_c}, \overrightarrow{A_s}) - \sum_{i=1}^{M} \alpha_{i,s} K(\overrightarrow{A_c}, \overrightarrow{A}_{b_i}) + d_s, \quad (3)$$

where $K(\cdot,\cdot)$ could be any sequence kernels that satisfy the Mercer condition, $\alpha_{0,s}$ is the Lagrange multiplier corresponding to the target speaker, $\alpha_{i,s}$ $(i = 1, \ldots, M)$ are Lagrange multipliers (some of which may be zero) corresponding to the background speakers, and d_s is a bias term.

Comparing Eq. 2 and Eq. 3 and comparing Fig. 1 and Fig. 2 suggest that kernel-based SVM scoring is more general and is potentially better than LR scoring in two aspects. First, the SVM optimally selects the most appropriate background speakers through the non-zero $\alpha_{i,s}$. Second, instead of using a single background model that contains the average characteristics of all background speakers, a specific set of background speakers is used for each target speaker for scoring. This is to some extent analogous to cohort scoring [26]. However, the cohort set is now discriminatively and optimally determined by SVM training, and the contribution of the selected background models is also optimally weighted through the Lagrange multipliers $\alpha_{i,s}$.

III. SIMILARITY METRICS AND SEQUENCE KERNELS

Comparing Fig. 1 and Fig. 2 and comparing Eq. 2 and Eq. 3 suggest that the sequence kernels $K(\overrightarrow{A_c}, \overrightarrow{A_s})$ in Eq. 2 can be derived from a similarity metric or similarity function $f(\overrightarrow{A_c}, \overrightarrow{A_s})$. However, to make sure that the SVM training algorithm converges to a stable solution, the function f inside the circle in Fig. 1 should satisfy the Mercer condition [27], i.e., $f(\overrightarrow{A_c}, \overrightarrow{A_s})$ can be expressed as $\left\langle \phi(\overrightarrow{A_c}), \phi(\overrightarrow{A_s}) \right\rangle$. For those similarity functions that do not satisfy this requirement, e.g. $f\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) = \left\langle \overrightarrow{A_c}, \log \overrightarrow{A_s}/\overrightarrow{A_b} \right\rangle$ in Table I, some approximations will need to be made. The following subsections describe four commonly used kernels derived from a specific similarity metric or function. Table II summarizes the properties of these kernels.

TABLE II

SEQUENCE KERNELS $K(\overrightarrow{A_c}, \overrightarrow{A_s}) = \left\langle \boldsymbol{Q}^{-\frac{1}{2}} \overrightarrow{A}_c, \boldsymbol{Q}^{-\frac{1}{2}} \overrightarrow{A}_s \right\rangle$ AND THEIR CORRESPONDING SIMILARITY METRICS.

Kernel	Specific Similarity Metric	Matrix $\boldsymbol{Q}$
Euclidean	Euclidean Distance	$\boldsymbol{I}$
GSV [18]	KL Divergence	$\boldsymbol{Q}_{\text{GSV}}$ (Eq. 6)
LLR [17]	KL Divergence	$\text{diag}\left\{\overrightarrow{A}_b\right\}$
GLDS [15]	Linear Discriminant	$\frac{1}{M}\sum_{i=1}^{M} \overrightarrow{A}_{b_i} \overrightarrow{A}_{b_i}^{\text{T}}$

A. Euclidean Kernel

The simplest type of Mercer kernel is a linear kernel:

$$K_{\text{E}}(\overrightarrow{A_c}, \overrightarrow{A_s}) = \left\langle \overrightarrow{A_c}, \overrightarrow{A_s} \right\rangle. \quad (4)$$

Note that this kernel can be obtained from the Euclidean distance between vectors in the feature space [27]. Therefore, we refer to it as Euclidean kernel.

B. Divergence Kernel

One commonly used distance metric for probability distributions is the Kullback-Leibler (KL) divergence. Here, we highlight two types of divergence kernels: GMM-supervector kernel and linearized likelihood-ratio kernel.

1) GMM-Supervector (GSV) kernel: Campbell et al. [9] use the log-sum inequality to approximate the KL divergence between two GMMs with the same mixture weights $\lambda_{b,i}$ and covariance matrices $\boldsymbol{\Sigma}_{b,i}$ but with different mean vectors ($\boldsymbol{\mu}_{c,i}$ and $\boldsymbol{\mu}_{s,i}$). The approximation leads to the GMM-supervector kernel:

$$K_{\text{GSV}}(\overrightarrow{A_c}, \overrightarrow{A_s}) = \overrightarrow{A_c}^{\text{T}} \boldsymbol{Q}_{\text{GSV}}^{-1} \overrightarrow{A_s} = \left\langle \boldsymbol{Q}_{\text{GSV}}^{-\frac{1}{2}} \overrightarrow{A_c}, \boldsymbol{Q}_{\text{GSV}}^{-\frac{1}{2}} \overrightarrow{A_s} \right\rangle, \quad (5)$$

where

$$\boldsymbol{Q}_{\text{GSV}} = \text{diag}\left\{\lambda_{b,1}^{-1} \text{diag}(\boldsymbol{\Sigma}_{b,1}), \ldots, \lambda_{b,G}^{-1} \text{diag}(\boldsymbol{\Sigma}_{b,G})\right\}. \quad (6)$$

2) Linearized Likelihood-Ratio Kernel: Kernels for discrete models such as n-grams [16], [17] can be derived from the similarity measure between the claimant (test) model $\overrightarrow{A_c}$ and the target-speaker model $\overrightarrow{A_s}$ using the KL divergence:

$$\begin{aligned} f\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) &= \mathcal{D}_{\text{KL}}(\overrightarrow{A_c} \parallel \overrightarrow{A_b}) - \mathcal{D}_{\text{KL}}(\overrightarrow{A_c} \parallel \overrightarrow{A_s}) \\ &= \left\langle \overrightarrow{A_c}, \log \frac{\overrightarrow{A_c}}{\overrightarrow{A_b}} \right\rangle - \left\langle \overrightarrow{A_c}, \log \frac{\overrightarrow{A_c}}{\overrightarrow{A_s}} \right\rangle = \left\langle \overrightarrow{A_c}, \log \frac{\overrightarrow{A_s}}{\overrightarrow{A_b}} \right\rangle, \end{aligned} \tag{7}$$

where the notation $\log \frac{\overrightarrow{X}}{\overrightarrow{Y}}$ means element-wise division followed by logarithm. Using the technique in [17] to approximate the KL divergence, the linearlized likelihood ratio kernel can be obtained:

$$K_{\text{LR}}(\overrightarrow{A_c}, \overrightarrow{A_s}) = \overrightarrow{A_c}^{\text{T}} \boldsymbol{Q}_{\text{LR}}^{-1} \overrightarrow{A_s} = \left\langle \boldsymbol{Q}_{\text{LR}}^{-\frac{1}{2}} \overrightarrow{A_c}, \boldsymbol{Q}_{\text{LR}}^{-\frac{1}{2}} \overrightarrow{A_s} \right\rangle \tag{8}$$

where $\boldsymbol{Q}_{\text{LR}} = \text{diag}\left\{\overrightarrow{A_b}\right\} = \text{diag}\left\{A_{b,1}, \ldots, A_{b,i}, \ldots, A_{b,N}\right\}$.

C. GLDS Kernel

A kernel can be obtained by finding a linear disclaimant function $f_s(\overrightarrow{A_c}) = \mathbf{w}_s^{\text{T}} \overrightarrow{A_c}$ that optimally divides the training data into target-speaker class and impostor class. This leads to the generalized linear discriminant sequence (GLDS) kernel [15]:

$$K_{\text{GLDS}}(\overrightarrow{A_c}, \overrightarrow{A_s}) = \left\langle \boldsymbol{Q}_{\text{GLDS}}{}^{-\frac{1}{2}} \overrightarrow{A}_c, \boldsymbol{Q}_{\text{GLDS}}{}^{-\frac{1}{2}} \overrightarrow{A}_s \right\rangle \tag{9}$$

where $\boldsymbol{Q}_{\text{GLDS}} = \frac{1}{M} \sum_{k=1}^{M} \overrightarrow{A}_{b_k} \overrightarrow{A}_{b_k}^{\text{T}}$ is a second moment matrix of $\overrightarrow{A}_{b_i}$ derived from background speakers.

IV. Optimization of Kernels

A common characteristic of the kernels in Section III is that they are all derived under the assumption that the discriminant function or similarity metric has a specific form. For example, the GSV kernel is derived from KL divergence, the linearized LR kernel is derived from discriminant function $f_s(\overrightarrow{A_c}) = \left\langle \overrightarrow{A_c}, \log \overrightarrow{A_s}/\overrightarrow{A_b} \right\rangle$, and the GLDS kernel is derived from linear discriminant function $f_s(\overrightarrow{A_c}) = \mathbf{w}_s^{\text{T}} \overrightarrow{A_c}$. This constraint can be relaxed by using a general discriminant function $f_s(\overrightarrow{A})$. This section derives two new kernels, namely regression optimized kernel and maximum-margin empirical kernel, based on two different approaches to optimizing a general discriminant function.

A. Regression Optimized Kernel

1) Formulation: Instead of assuming a specific form for the discriminant function as in the GLDS kernel, our derivation begins with a general discriminant function: $f(\overrightarrow{A}, \overrightarrow{A_s}) \triangleq f_s(\overrightarrow{A})$. Our goal is to derive a kernel from the "best" discriminant function $\hat{f}_s(\overrightarrow{A})$ that optimally divides the training data into $\{\overrightarrow{A_s};\ y_s = +1\}$ and $\{\overrightarrow{A_{b_k}};\ y_{b_k} = 0\}_{k=1}^{M}$.[2] This can be achieved by solving:

$$\hat{f}_s = \arg\min_{f_s \in \mathcal{H}} \left\{ \sum_{i \in \{s, b_k\}_{k=1}^{M}} \gamma_i L\left(f_s(\overrightarrow{A_i}), y_i\right) + \lambda \parallel f_s \parallel^2 \right\} \tag{10}$$

where M is the number of background speakers, $\lambda > 0$ is a regularizing parameter, $L(\cdot, \cdot)$ is a loss function, and γ_i is to alleviate the imbalance between the two classes of data. According to [28], the optimal solution of Eq. 10 can be written as:

$$\hat{f}_s(\overrightarrow{A}) = \sum_{i \in \{s, b_k\}_{k=1}^{M}} w_{s,i} k(\overrightarrow{A}, \overrightarrow{A_i}), \tag{11}$$

where $w_{s,i}$ are speaker-dependent weights and $k(\cdot, \overrightarrow{A_i}) : \mathbb{R}^N \times \mathbb{R}^N \mapsto \mathbb{R}$ are kernels in the reproducing kernel Hilbert space $\mathcal{H}$ such that

$$\left\langle f_s(\cdot), k(\cdot, \overrightarrow{A_i}) \right\rangle_{\mathcal{H}} = f_s(\overrightarrow{A_i}) \qquad \forall\ f_s \in \mathcal{H}. \tag{12}$$

When $L(\cdot, \cdot)$ is a squared loss function, the optimization problem amounts to finding the combination weights $w_{s,i}$ for which regression analysis using the least squares method is a natural solution. Eq. 11 suggests that supervector $\overrightarrow{A}$ is first mapped to an $(M+1)$-dim space defined by $k(\cdot, \overrightarrow{A_i})$. Regression analysis is then performed in this space.

Eq. 11 and Eq. 12 suggest that

$$\begin{aligned} \parallel \hat{f}_s \parallel^2 &= \langle \hat{f}_s, \hat{f}_s \rangle = \left\langle \hat{f}_s, \sum_{i \in \{s, b_k\}_{k=1}^{M}} w_{s,i} k(\overrightarrow{A_i}, \cdot) \right\rangle \\ &= \sum_{i \in \{s, b_k\}_{k=1}^{M}} w_{s,i} \left(\sum_{j \in \{s, b_k\}_{k=1}^{M}} w_{s,j} k(\overrightarrow{A_i}, \overrightarrow{A_j}) \right) \\ &= \mathbf{w}_s^{\text{T}} \mathbf{K}_s \mathbf{w}_s, \end{aligned} \tag{13}$$

where $\mathbf{w}_s = [w_{s,s}, w_{s,b_1}, \ldots, w_{s,b_M}]^{\text{T}}$ and

$$\mathbf{K}_s = \begin{bmatrix} k_{s,s} & k_{b_1,s} & \cdots & k_{b_M,s} \\ k_{s,b_1} & k_{b_1,b_1} & \cdots & k_{b_M,b_1} \\ \vdots & \vdots & \ddots & \vdots \\ k_{s,b_M} & k_{b_1,b_M} & \cdots & k_{b_M,b_M} \end{bmatrix}, \tag{14}$$

where $k_{i,j} = k_{j,i} = k(\overrightarrow{A_i}, \overrightarrow{A_j})$.

Therefore, the optimization problem in Eq. 10 can be formulated as:

$$\min_{\mathbf{w}_s \in \mathbb{R}^{M+1}} \ \left\{ (\mathbf{y} - \mathbf{K}_s \mathbf{w}_s)^{\text{T}} \boldsymbol{\Gamma} (\mathbf{y} - \mathbf{K}_s \mathbf{w}_s) + \lambda \mathbf{w}_s{}^{\text{T}} \mathbf{K}_s \mathbf{w}_s \right\} \tag{15}$$

where $\mathbf{y} = [1, 0, \ldots, 0]_{(M+1)\times 1}^{\text{T}}$,

$$\boldsymbol{\Gamma} = \text{diag}\{\gamma_s, \gamma_{b_1}, \ldots, \gamma_{b_M}\} = \text{diag}\{\gamma^+, \gamma^-, \ldots, \gamma^-\}. \tag{16}$$

Setting the derivative of the objective function in Eq. 15 to zero, we obtain the optimal value of $\mathbf{w}_s$:

$$\mathbf{w}_s = (\mathbf{K}_s \boldsymbol{\Gamma} \mathbf{K}_s + \lambda \mathbf{K}_s)^{-1} (\mathbf{K}_s \boldsymbol{\Gamma} \mathbf{y}), \tag{17}$$

where we have used the symmetric property of $\mathbf{K}_s$.

[2] Setting the ideal outputs as $y_s = +1$ and $y_{b_k} = 0$ (instead of -1) will greatly simplify subsequence derivation.

Using Eqs. 16–17, we can express the optimal discriminant function (Eq. 11) as:

$$\hat{f}_s(\overrightarrow{A}) = \sum_{i\in\{s,b_k\}_{k=1}^{M}} w_{s,i}k(\overrightarrow{A},\overrightarrow{A_i})$$

$$= \left[(\mathbf{K}_s\mathbf{\Gamma}\mathbf{K}_s + \boldsymbol{\lambda}\mathbf{K}_s)^{-1}(\mathbf{K}_s\mathbf{\Gamma}\mathbf{y})\right]^{\mathrm{T}}_{(M+1)\times 1} \begin{bmatrix} k(\overrightarrow{A},\overrightarrow{A_s}) \\ k(\overrightarrow{A},\overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A},\overrightarrow{A_{b_M}}) \end{bmatrix}$$

$$= \gamma^{+} \begin{bmatrix} k(\overrightarrow{A_s},\overrightarrow{A_s}) \\ k(\overrightarrow{A_s},\overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_s},\overrightarrow{A_{b_M}}) \end{bmatrix}^{\mathrm{T}} (\mathbf{K}_s\mathbf{\Gamma}\mathbf{K}_s + \lambda\mathbf{K}_s)^{-1} \begin{bmatrix} k(\overrightarrow{A},\overrightarrow{A_s}) \\ k(\overrightarrow{A},\overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A},\overrightarrow{A_{b_M}}) \end{bmatrix}.$$

Because γ^{+} is a constant, it can be discarded without affecting the discriminative ability of $\hat{f}_s(\overrightarrow{A})$. Note that the matrix $\mathbf{K}_s$ and the vector $k(\overrightarrow{A},\cdot)|_{(s,b_1,\ldots,b_M)}$ are target speaker-dependent.[3] Consider that these matrices and vectors are dominated by non-target speaker data; to make $f_s(\overrightarrow{A}_c)$ symmetric and to reduce computation time and storage space, we perform the following approximations:

$$\mathbf{K}_s \doteq \mathbf{K} = \begin{bmatrix} k_{b,b} & k_{b_1,b} & \cdots & k_{b_M,b} \\ k_{b,b_1} & k_{b_1,b_1} & \cdots & k_{b_M,b_1} \\ \vdots & \vdots & \ddots & \vdots \\ k_{b,b_M} & k_{b_1,b_M} & \cdots & k_{b_M,b_M} \end{bmatrix}, \quad (18)$$

and

$$k(\overrightarrow{A},\cdot)|_{(s,b_1,\ldots,b_M)} \doteq k(\overrightarrow{A},\cdot)|_{(b,b_1,\ldots,b_M)},$$

where the universal background supervector $\overrightarrow{A_b}$ is used to approximate $\overrightarrow{A_s}$. These approximations are valid because when the number of background speakers M is sufficiently large, small variation in one component of the $(M+1)$-dim vector $k(\overrightarrow{A},\cdot)|_{(s,b_1,\ldots,b_M)}$ will not cause it to deviate from its true position significantly. Moreover, as $\overrightarrow{A_s}$ is adapted from $\overrightarrow{A_b}$, the closest approximation to $\overrightarrow{A_s}$ is $\overrightarrow{A_b}$.

With the above approximations, the regression optimized kernel is written as:

$$K_{\mathrm{Reg}}(\overrightarrow{A_c},\overrightarrow{A_s}) = \Big\langle (\mathbf{K}\mathbf{\Gamma}\mathbf{K} + \lambda\mathbf{K})^{-\frac{1}{2}} k(\overrightarrow{A_c},\cdot)|_{(b,b_1,\ldots,b_M)}, (\mathbf{K}\mathbf{\Gamma}\mathbf{K} + \lambda\mathbf{K})^{-\frac{1}{2}} k(\overrightarrow{A_s},\cdot)|_{(b,b_1,\ldots,b_M)} \Big\rangle, \quad (19)$$

where $\mathbf{K}$ and $\mathbf{\Gamma}$ are defined in Eqs. 18 and 16. $(\mathbf{K}\mathbf{\Gamma}\mathbf{K} + \lambda\mathbf{K})^{-\frac{1}{2}}$ can be considered as a normalization matrix computed from the background speakers. Note that $k_{i,j} = k(\overrightarrow{A_i},\overrightarrow{A_j})$ should belong to $\mathcal{H}$. For low-level systems, one possibility is to use the GSV kernel; for high-level systems, the linearized LR kernel can be used.

2) Double-Optimization Procedure: The above derivation suggests that constructing a regression-kernel-based SVM for target speaker s involves a 2-step optimization process:

Step 1 Find the weights $w_{s,i}$ in Eq. 11 that optimize the objective function in Eq. 10, which leads to the normalization matrix $\mathbf{K}\mathbf{\Gamma}\mathbf{K} + \lambda\mathbf{K}$ in Eq. 19.

Step 2 Optimize the Lagrange multipliers in Eq. 3 via SVM training using Eq. 19 as the kernel, with $\overrightarrow{A_c}$ replaced by $\overrightarrow{A_s}$ for speaker-class data and by $\overrightarrow{A_{b_j}}$, $j = 1,\ldots,M$, for impostor-class data.

During verification, given a test utterance, a supervector $\overrightarrow{A_c}$ is derived and is applied to Eq. 3 to compute the verification score, using Eq. 19 as the kernel. Fig. 3 illustrates the scoring process and the structure of the regression optimized kernel.

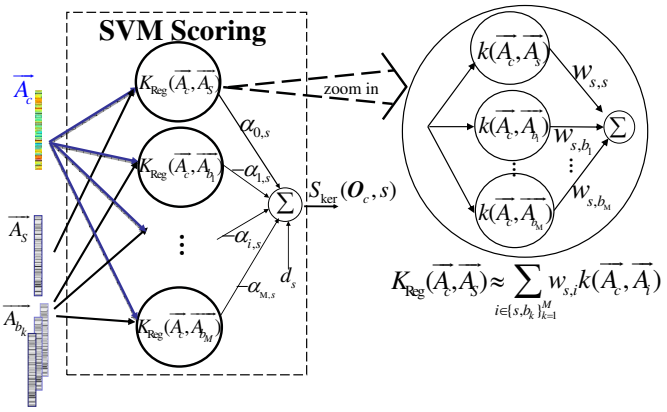

Fig. 3. The scoring process of the regression optimized kernel. $\alpha_{i,s}$ and $w_{s,i}$ (in red) are optimized by a double-optimization procedure described in Section IV-A2.

B. Maximum-Margin Empirical Kernel

1) Formulation: If the lost function $L(x,y)$ in Eq. 10 is the Vapnik's ϵ-insensitive loss function [20]

$$L(x,y) = \begin{cases} 0 & \text{if } |x-y| < \epsilon \\ |x-y| - \epsilon & \text{otherwise,} \end{cases}$$

then it can be shown [28] that the minimization in Eq. 10 is equivalent to the SVM training algorithm. Therefore, we can generalize Eq. 11 to

$$f_s(\overrightarrow{A}) = v_{s,0}k(\overrightarrow{A},\overrightarrow{A_s}) - \sum_{i\in\mathcal{S}_b} v_{s,i}k(\overrightarrow{A},\overrightarrow{A_i}) + d'_s, \quad (20)$$

where $\mathcal{S}_b \subseteq \{b_k\}_{k=1}^{M}$ is a set of support vector indexes from the negative class, $v_{s,0}$ is the Lagrange multiplier corresponding to the (solely) positive support vector,[4] and $v_{s,i}$ are the Lagrange multipliers corresponding to the negative support vectors. Therefore, the optimal weights (Lagrange multipliers and bias) in Eq. 20 can be found by maximizing the margin of an SVM that separates the target speaker s from background speakers b_k. We cannot, however, use Eq. 20 as a kernel, because it may not satisfy the Mercer's condition. One possible solution is to use empirical kernel map [21]–[23] as follows.

Assume that we have M background speakers. We first train a UBM using these M speakers, which results in a supervector denoted $\overrightarrow{A_b}$. For the i-th background speaker, an SVM is trained to distinguish his/her voice from that of the other $M-1$ background speakers and the UBM. Similarly, an SVM is trained to distinguish the UBM from all of the M

[3] $k(\overrightarrow{A},\cdot)|_{(s,b_1,\ldots,b_M)} \triangleq \left[k(\overrightarrow{A},\overrightarrow{A_s}), k(\overrightarrow{A},\overrightarrow{A_{b_1}}), \ldots, k(\overrightarrow{A},\overrightarrow{A_{b_M}})\right]^{\mathrm{T}}$.

[4] We assume that each target speaker has one enrollment utterance. Generalization to multiple training utterances is trivial.

background speakers. Denote the output of the i-th SVM as $f_{b_i}(\overrightarrow{A})$ and that corresponding to the UBM as $f_b(\overrightarrow{A})$, where we have replaced s in Eq. 20 by b_i and b. During enrollment, given an utterance from a target speaker s, we estimate the corresponding supervector $\overrightarrow{A_s}$ and present it to the UBM's SVM and M background SVMs. We also present the UBM's supervector $\overrightarrow{A_b}$ and each of the background supervectors $\overrightarrow{A_{b_i}}$ to the speaker's SVM. The two sets of outputs are averaged to produce an $(M+1)$-dim vector:

$$\mathbf{f}_s = \frac{1}{2}\begin{bmatrix} f_b(\overrightarrow{A_s}) + f_s(\overrightarrow{A_b}) \\ f_{b_1}(\overrightarrow{A_s}) + f_s(\overrightarrow{A_{b_1}}) \\ \cdots \\ f_{b_M}(\overrightarrow{A_s}) + f_s(\overrightarrow{A_{b_M}}) \end{bmatrix}.$$

This vector represents the speaker class for training a linear scoring SVM. Vectors representing the impostor class are obtained by presenting each of the background speakers to the UBM's SVM and the M background SVMs, which results in M training vectors:

$$\mathbf{f}_{b_j} = \frac{1}{2}\begin{bmatrix} f_b(\overrightarrow{A_{b_j}}) + f_{b_j}(\overrightarrow{A_b}) \\ f_{b_1}(\overrightarrow{A_{b_j}}) + f_{b_j}(\overrightarrow{A_{b_1}}) \\ \cdots \\ f_{b_M}(\overrightarrow{A_{b_j}}) + f_{b_j}(\overrightarrow{A_{b_M}}) \end{bmatrix},\ j = 1, \ldots, M.$$

The kernel of the scoring SVM is given by

$$K_{\text{MM-Emp}}(\overrightarrow{A_c}, \overrightarrow{A_s}) = < \mathbf{F}_b^{-\frac{1}{2}}\mathbf{f}_c, \mathbf{F}_b^{-\frac{1}{2}}\mathbf{f}_s >, \qquad (21)$$

where $\mathbf{F}_b = [\mathbf{f}_b \ \ \mathbf{f}_{b_1} \ \ \ldots \ \ \mathbf{f}_{b_M}]$. We refer to $K_{\text{MM-Emp}}$ as the maximum-margin empirical kernel.

2) Double-Optimization Procedure: Again, constructing a maximum-margin kernel-based SVM for target speaker s involves a 2-step optimization process:

Step 1 Optimize the Lagrange multipliers $v_{s,i}$ in Eq. 20 via SVM training, using $k(\cdot,\cdot)$ as the kernel. For low-level systems, $k(\cdot,\cdot)$ is the GSV kernel (Eq. 5), and for high-level systems, $k(\cdot,\cdot)$ is the linearized LR kernel (Eq. 8).

Step 2 Optimize the Lagrange multipliers $\alpha_{s,i}$ in Eq. 3 via SVM training using Eq. 21 as the kernel, with $\mathbf{f}_c$ replaced by $\mathbf{f}_s$ for speaker-class data and by $\mathbf{f}_{b_j}$, $j = 1, \ldots, M$, for impostor-class data.

3) Advantages: The maximum-margin empirical kernel has two advantages over the regression optimized kernel.

- *Ease of Training*. Apart from the penalty factor in SVM training, no parameters need to be tuned. The training of the regression optimized kernel, on the other hand, requires tuning the parameters $\mathbf{\Gamma}$ and λ in Eq. 19.
- *Avoid the Inference of Outliers*. In the regression optimized kernel, supervectors that are mapped to points far away from the decision plane defined by $\{v_{s,i}\}$ in the $(M+1)$-dim space will have significant influence on the position and orientation of the plane, which may not be desirable. On the other hand, the SVM training algorithm will pick the supervectors that are mapped to points close to the decision plane as support vectors, thereby allowing these important vectors to have a higher influence on the decision plane.

V. Relationship Between Different Kernels

A. Regression Optimized Kernels Vs. Other Kernels

The regression optimized kernel can be considered as a general form of the Euclidean, GSV, linearized LR and GLDS kernels. Starting from Eq. 19, if $\mathbf{\Gamma} = \mathbf{0}$ and $\lambda = 1$, then the (i,j)-th element of the regression optimized kernel matrix $\mathbf{K}_{\text{Reg}}$ becomes:

$$\begin{aligned} \{\mathbf{K}_{\text{Reg}}\}_{i,j} &= K_{\text{Reg}}(\overrightarrow{A_i}, \overrightarrow{A_j}) \\ &= \left\langle \mathbf{K}^{-\frac{1}{2}} \begin{bmatrix} k(\overrightarrow{A_i}, \overrightarrow{A_b}) \\ k(\overrightarrow{A_i}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_i}, \overrightarrow{A_{b_M}}) \end{bmatrix}, \mathbf{K}^{-\frac{1}{2}} \begin{bmatrix} k(\overrightarrow{A_j}, \overrightarrow{A_b}) \\ k(\overrightarrow{A_j}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_j}, \overrightarrow{A_{b_M}}) \end{bmatrix} \right\rangle \\ &= \left\langle \phi(\overrightarrow{A_i}), \phi(\overrightarrow{A_j}) \right\rangle. \end{aligned} \qquad (22)$$

Define $\mathbf{\Omega}_s = \left[\phi(\overrightarrow{A_s}), \phi(\overrightarrow{A_{b_1}}), \ldots, \phi(\overrightarrow{A_{b_M}})\right]$. Then we have

$$\mathbf{\Omega}_s = \mathbf{K}^{-\frac{1}{2}} \begin{bmatrix} k_{s,b} & k_{b_1,b} & \cdots & k_{b_M,b} \\ k_{s,b_1} & k_{b_1,b_1} & \cdots & k_{b_M,b_1} \\ \vdots & \vdots & \ddots & \vdots \\ k_{s,b_M} & k_{b_1,b_M} & \cdots & k_{b_M,b_M} \end{bmatrix} \doteq \mathbf{K}^{-\frac{1}{2}}\mathbf{K}_s,$$

where $\mathbf{K}_s$ is defined in Eq. 14. Therefore, using Eq. 22, the regression optimized kernel matrix for target speaker s is:

$$\begin{aligned} \mathbf{K}_{\text{Reg}}^s &= \mathbf{\Omega}_s^{\text{T}}\mathbf{\Omega}_s = (\mathbf{K}^{-\frac{1}{2}}\mathbf{K}_s)^{\text{T}}(\mathbf{K}^{-\frac{1}{2}}\mathbf{K}_s) \\ &= \mathbf{K}_s^{\text{T}}\mathbf{K}^{-\frac{1}{2}}\mathbf{K}^{-\frac{1}{2}}\mathbf{K}_s \doteq \mathbf{K}_s. \quad \text{(because Eq. 18: } \mathbf{K} \doteq \mathbf{K}_s) \end{aligned}$$

Consider the elements of $\mathbf{K}_s$. If we choose

$$\begin{aligned} k_{i,j} &= k(\overrightarrow{A_i}, \overrightarrow{A_j}) = K_{\text{GSV}}(\overrightarrow{A_i}, \overrightarrow{A_j}) \\ &= \sum_{g=1}^{G} \left(\sqrt{\lambda_{b,g}}\mathbf{\Sigma}_{b,g}^{-\frac{1}{2}}\boldsymbol{\mu}_{i,g}\right)^{\text{T}} \left(\sqrt{\lambda_{b,g}}\mathbf{\Sigma}_{b,g}^{-\frac{1}{2}}\boldsymbol{\mu}_{j,g}\right), \end{aligned} \qquad (23)$$

then the regression optimized kernel matrix $\mathbf{K}_{\text{Reg}}^s$ becomes the GSV kernel matrix $\mathbf{K}_{\text{GSV}}^s$. The above derivation can be generalized to other kernels. Therefore, by choosing special values of $\mathbf{\Gamma}$ and λ and by using a special form of $k(\overrightarrow{A_i}, \overrightarrow{A_j})$ in Eq. 19, the regression optimized kernel can be reduced to other sequence kernels.

This generalization property can also be observed from the scoring procedure shown in Fig. 3. For example, if the number of inner nodes in Fig. 3 reduces to one per outer node, then regression kernel scoring reduces to Euclidean, GSV, linearized LR or GLDS kernel scoring. Further, if the number of outer nodes in Fig. 3 reduces to two with $\alpha_{0,s} = \alpha_{1,s} = 1$, then kernel scoring reduces to LR scoring.

The discriminant function (Eq. 11) that leads to the regression kernel has a form similar to the sparse multiple-kernel [29]. However, there are two major differences. First, in Eq. 11 the number of kernels is equal to the number of training vectors, whereas in the sparse multiple-kernel the number of kernels is pre-defined. Second, the weights $w_{s,i}$ in Eq. 11 is obtained by linear regression whereas the combination weights in the sparse multiple-kernel are estimated by gradient projection.

B. Regression Optimized Kernel Vs. Maximum-Margin Empirical Kernel

In Eq. 19, when $\mathbf{\Gamma} = \mathbf{0}$ and $\lambda = 1$, the regression optimized kernel becomes:

$$K_{\text{Reg}}(\overrightarrow{A_c}, \overrightarrow{A_s}) = \left\langle \mathbf{K}^{-\frac{1}{2}} \begin{bmatrix} k(\overrightarrow{A_c}, \overrightarrow{A_b}) \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_c}, \overrightarrow{A_{b_M}}) \end{bmatrix}, \mathbf{K}^{-\frac{1}{2}} \begin{bmatrix} k(\overrightarrow{A_s}, \overrightarrow{A_b}) \\ k(\overrightarrow{A_s}, \overrightarrow{A_{b_1}}) \\ \vdots \\ k(\overrightarrow{A_s}, \overrightarrow{A_{b_M}}) \end{bmatrix} \right\rangle. \quad (24)$$

Rewrite the maximum-margin empirical kernel in Eq. 21 as Eq. 25. Comparing Eq. 24 and Eq. 25 suggests that the maximum-margin empirical kernel is a general form of the regression optimized kernel.

C. Comparing Computational Complexity

Table III shows the computational complexity of different kernels. In terms of scoring, the Euclidean distance, GSV and linearized LR kernels are the least complex, while the maximum-margin empirical kernel are the most complex.

VI. Experiments

The kernels in Sections III and IV were used for high- and low-level speaker verification. This section describes the speech data and evaluation procedures.

A. Low-Level Speaker Verification

The classical GMM-UBM [2] and GMM-SVM [18] were used as the baselines for comparison. For the GMM-UBM, gender-dependent UBMs with 1,024 Gaussians were used, because in NIST speaker recognition evaluation, each hypothesized speaker will only be tested against utterances of the same gender. The GMMs of target speakers were adapted from the UBMs using MAP adaptation [2], with relevance factor $r = 16$. Each supervector in the GMM-SVM comprises the mean vectors of a MAP-adapted GMM, each with 256 Gaussians.[5]

For each utterance, an energy-based voice activity detector was used to remove the silence regions. Twelfth-order MFCCs [3] plus their first derivative were extracted from the speech regions of the utterance using a 25-ms Hamming window with a shift of 10 ms, leading to a 24-dim acoustic vector per frame. Cepstral mean normalization [30] was applied to the MFCCs, followed by feature warping [31].

To reduce the effect of session variability, nuisance attribute projection (NAP) [9] with corank=8 was applied to the supervectors. The NAP parameters were obtained from speakers in NIST SRE 2001 [32] who provide multiple conversations in different sessions. This amounts to 74 male speakers and 100 female speakers, each providing 12 conversations on average.

T-norms [7] were applied to normalize the SVM scores and the LR scores to further reduce the effect of session variability.

[5]We have tried using different numbers of Gaussians and found that 256 gives the best performance.

B. High-Level Speaker Verification

We used articulatory features (AFs) to build speaker-dependent pronunciation models. AFs are representations describing the movements or positions of different articulators during speech production. Following [14], [33], we used 6 manner classes and 10 place classes to describe the articulators. AFs were automatically determined from speech signals using AF-based multilayer perceptrons (MLPs) [33]. More specifically, given a sequence of acoustic vectors (MFCCs) $\mathbf{x}_t$ where $t = 1, \ldots, T$, the MLPs produce a sequence of manner and place labels. These labels were then used to create an AF-based conditional pronunciation model (AFCPM) using MAP adaptation [34]. Each AFCPM comprises the joint densities of 6 manner and 10 place classes, conditioned on 12 phonetic classes, leading to a 720-dim AF-supervector. T-norm was applied during scoring.

The phone recognizer for extracting AFs uses standard 39-dim vectors comprising MFCCs, energy, and their derivatives. The inputs to the manner and place MLP comprise 9 frames of 26-dim acoustic vectors: 12 MFCCs, log-energy, and their first derivatives.

C. Speech Corpora

NIST SRE 2001 [32], NIST SRE 2002 [35], SPIDRE [36], and HTIMIT [37] were used in the experiments.

NIST SRE'01 contains 2,350 cellular-phone conversations extracted from the Switchboard-II Phase IV Corpus. All of the utterances in NIST01 were used for creating the background models. All of the training utterances (112 male utterances and 122 female utterances) in the corpus were used as gender-dependent impostor data when training the target-speaker SVMs. Test utterances with length (after silence removal) longer than 25 seconds were used for creating the T-norm speaker models, which amount to 127 male and 145 female T-norm speakers. The corpus was also used for computing the NAP projection matrix. Specifically, speakers with multiple conversations were identified and the conversations of these speakers are assumed to be extracted from different sessions. This amounts to 74 male speakers and 100 female speakers, each providing 12 conversations on average.

NIST SRE'02 contains the cellular-phone conversations of 139 male and 191 female target speakers. All of these speakers were used in the evaluation. We followed the protocol of one-speaker detection task, which amounts to 2,983 true-speaker trials and 36,287 impostor attacks.

HTIMIT and SPIDRE were used to train the MLPs and the phone recognizer for high-level speaker verification. Specifically, 3,794 utterances selected from HTIMIT were used to train the manner and place MLPs, and utterances from SPIDRE were used to train a null-grammar phoneme recognizer with 46 context-independent phoneme models (HMMs with 3 states, 16 mixtures per state).

D. Training of SVMs and Kernels

The SVM of each target speaker was trained by using his/her training utterance as the positive sample and the

$$K_{\text{MM-Emp}}(\overrightarrow{A_c},\overrightarrow{A_s}) = \left\langle \mathbf{F}_b^{-\frac{1}{2}} \begin{bmatrix} (v_{b,0}+v_{c,0})k(\overrightarrow{A_c},\overrightarrow{A_b}) - \sum_i v_{b,i}k(\overrightarrow{A_c},\overrightarrow{A_i}) - \sum_i v_{c,i}k(\overrightarrow{A_b},\overrightarrow{A_i}) + d'_b + d'_c \\ (v_{b_1,0}+v_{c,0})k(\overrightarrow{A_c},\overrightarrow{A_{b_1}}) - \sum_i v_{b_1,i}k(\overrightarrow{A_c},\overrightarrow{A_i}) - \sum_i v_{c,i}k(\overrightarrow{A_{b_1}},\overrightarrow{A_i}) + d'_{b_1} + d'_c \\ \vdots \\ (v_{b_M,0}+v_{c,0})k(\overrightarrow{A_c},\overrightarrow{A_{b_M}}) - \sum_i v_{b_M,i}k(\overrightarrow{A_c},\overrightarrow{A_i}) - \sum_i v_{c,i}k(\overrightarrow{A_{b_M}},\overrightarrow{A_i}) + d'_{b_M} + d'_c \end{bmatrix}, \right. \\ \left. \mathbf{F}_b^{-\frac{1}{2}} \begin{bmatrix} (v_{b,0}+v_{s,0})k(\overrightarrow{A_s},\overrightarrow{A_b}) - \sum_i v_{b,i}k(\overrightarrow{A_s},\overrightarrow{A_i}) - \sum_i v_{s,i}k(\overrightarrow{A_b},\overrightarrow{A_i}) + d'_b + d'_s \\ (v_{b_1,0}+v_{s,0})k(\overrightarrow{A_s},\overrightarrow{A_{b_1}}) - \sum_i v_{b_1,i}k(\overrightarrow{A_s},\overrightarrow{A_i}) - \sum_i v_{s,i}k(\overrightarrow{A_{b_1}},\overrightarrow{A_i}) + d'_{b_1} + d'_s \\ \vdots \\ (v_{b_M,0}+v_{s,0})k(\overrightarrow{A_s},\overrightarrow{A_{b_M}}) - \sum_i v_{b_M,i}k(\overrightarrow{A_s},\overrightarrow{A_i}) - \sum_i v_{s,i}k(\overrightarrow{A_{b_M}},\overrightarrow{A_i}) + d'_{b_M} + d'_s \end{bmatrix} \right\rangle \tag{25}$$

TABLE III
COMPUTATIONAL COMPLEXITY OF DIFFERENT KERNEL SCORING METHODS. D IS THE DIMENSIONALITY OF $\boldsymbol{o}_t$, N IS THE DIMENSIONALITY OF $\vec{A}$, M IS THE NUMBER OF BACKGROUND SPEAKERS, T IS THE NUMBER OF FRAMES (LABELS) IN THE VERIFICATION UTTERANCE. IT IS ASSUMED THAT ALL OF THE NORMALIZATION MATRICES $\boldsymbol{Q}$ AND $\mathbf{F}$ HAVE BEEN PRE-COMPUTED.

Scoring Method	Scoring Equation	Kernel Function	Scoring Complexity
SVM Scoring	$\alpha_{0,s}K_{\text{ker}}(\overrightarrow{A_c},\overrightarrow{A_s}) - \sum_{i=1}^{M}\alpha_{i,s}K_{\text{ker}}(\overrightarrow{A_c},\overrightarrow{A}_{b_i}) + d_s$	$K_{\text{E}}\left(\overrightarrow{A_c},\overrightarrow{A_s}\right) = \left\langle \overrightarrow{A_c},\overrightarrow{A_s}\right\rangle$	$\mathcal{O}(N(M+1)+N^2T)$
		$K_{\text{GSV}}\left(\overrightarrow{A_c},\overrightarrow{A_s}\right) = \left\langle \boldsymbol{Q}_{\text{GSV}}{}^{-\frac{1}{2}}\overrightarrow{A}_c, \boldsymbol{Q}_{\text{GSV}}{}^{-\frac{1}{2}}\overrightarrow{A}_s\right\rangle$	$\mathcal{O}(N(M+1)+N^2T)$
		$K_{\text{LR}}\left(\overrightarrow{A_c},\overrightarrow{A_s}\right) = \left\langle \boldsymbol{Q}_{\text{LR}}{}^{-\frac{1}{2}}\overrightarrow{A}_c, \boldsymbol{Q}_{\text{LR}}{}^{-\frac{1}{2}}\overrightarrow{A}_s\right\rangle$	$\mathcal{O}(N(M+1)+N^2T)$
		$K_{\text{GLDS}}\left(\overrightarrow{A_c},\overrightarrow{A_s}\right) = \left\langle \boldsymbol{Q}_{\text{GLDS}}{}^{-\frac{1}{2}}\overrightarrow{A}_c, \boldsymbol{Q}_{\text{GLDS}}{}^{-\frac{1}{2}}\overrightarrow{A}_s\right\rangle$	$\mathcal{O}(N(M+1)+D^2T)$
		$K_{\text{Reg}}\left(\overrightarrow{A_c},\overrightarrow{A_s}\right) = k(\overrightarrow{A}_c,\cdot)^{\text{T}}(\mathbf{K\Gamma K}+\lambda\mathbf{K})^{-1}k(\overrightarrow{A}_s,\cdot)$	$\mathcal{O}((M+1)^2+N^2T+N(M+1))$
		$K_{\text{MM-Emp}}\left(\overrightarrow{A_c},\overrightarrow{A_s}\right) =< \mathbf{F}_b^{-\frac{1}{2}}\mathbf{f}_c, \mathbf{F}_b^{-\frac{1}{2}}\mathbf{f}_s >$	$\mathcal{O}(N^2T+N(M+1)^2+M^3)$
LR Scoring	$f(\overrightarrow{A_c},\overrightarrow{A_s}) - f(\overrightarrow{A_c},\overrightarrow{A_b}) + d_s$	—	$\mathcal{O}(N+N^2T)$
	$\frac{1}{T}\sum_t \log\left(p_s(\boldsymbol{o}_t)/p_b(\boldsymbol{o}_t)\right)$	—	$\mathcal{O}(NT)$

training utterances of the same gender in NIST SRE 2001 as negative training samples. This amounts to 112 male and 122 female negative samples for each SVM. The same set of data was used for training different types of kernels. The regression and maximum-margin kernels can leverage the double optimization process by using two different training sets for the two stages of optimization. However, to be fair to other kernels that only have one optimization step, we apply the same training set to the two optimization steps in the regression and maximum-margin kernels. SVMlight [38] was used for training the SVMs.

In Eq. 16, $\gamma^+ = \frac{M}{M+1}$ and $\gamma^- = \frac{1}{M+1}$, where M is the number of negative-class speakers in SVM training, i.e. 112 for male and 122 for female. Moreover, in Eq. 15, $\lambda = 0.8$ for high-level systems and $\lambda = 0.2$ for low-level systems. A small λ was chosen for low-level systems because their speaker models are more reliable; therefore less regularization is required. We did not attempt to optimize these parameters, although it may improve performance.

For high-level systems, we used the LR kernel (K_{LR}) as the reproducing kernel $k(\cdot,\cdot)$ in Eqs. 19 and 25. For low-level systems, we used the GMM-supervector kernel (K_{GSV}) as the reproducing kernel.

E. Fusion of High- and Low-Level Systems

The articulatory feature-based models and the acoustic GMMs characterize speakers at two different levels. The former represents the pronunciation behaviors of individual speakers, whereas the latter focuses on their vocal-tract characteristics. Therefore, fusing their scores is expected to improve speaker verification performance. In this work, the scores from articulatory feature-based models and the acoustic GMMs (GMM-UBM and GMM-SVM) were linearly combined to obtain the fused scores:

$$S_{\text{Fuse}}(\boldsymbol{O}) = \eta\frac{S_{\text{High}}(\boldsymbol{O})-\mu_{\text{High}}}{\sigma_{\text{High}}} + (1-\eta)\frac{S_{\text{Low}}(\boldsymbol{O})-\mu_{\text{Low}}}{\sigma_{\text{Low}}} \tag{26}$$

where μ and σ are the mean and standard deviation of scores, respectively.

VII. RESULTS AND DISCUSSIONS

Table IV shows the equal error rate (EER) and minimum decision cost (DCF) achieved by LR scoring and various types of kernel scoring in both low- and high-level speaker verification systems. Fig. 4 shows the detection error tradeoff (DET) performance for high-level systems. Fig. 5 show the corresponding performance for the low-level systems and the fusion of the high- and low-level systems.

A. LR Scoring Versus Kernel-based SVM Scoring

For low-level speaker verification, the performance of GSV kernel scoring (K_{GSV}) is better than that of LR scoring.

TABLE IV

PERFORMANCE (EER AND MINIMUM DCF) ACHIEVED BY DIFFERENT SCORING METHODS IN LOW-LEVEL AND HIGH-LEVEL SPEAKER VERIFICATION. IN LOW-LEVEL SYSTEMS, GLDS SUPERVECTORS ARE THE SECOND ORDER POLYNOMIAL EXPANSIONS [15] OF MFCCS. FOR OTHER KERNELS IN LOW-LEVEL SYSTEMS, SUPERVECTORS ARE THE STACKING OF THE GAUSSIANS MEAN VECTORS. IN HIGH-LEVEL SYSTEMS, SUPERVECTORS ARE FORMED BY STACKING THE ENTRIES IN THE PROBABILITY MASS FUNCTIONS (AFCPM [17]).

Scoring Method	Kernel Type	Formulation	EER		min. DCF	
			Low-L	High-L	Low-L	High-L
Kernel	Euclidean	$K_{\text{E}}\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) = \left\langle \overrightarrow{A_c}, \overrightarrow{A_s} \right\rangle$	12.57%	26.86%	0.0475	0.0944
	GSV	$K_{\text{GSV}}\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) = \left\langle \mathbf{Q}_{\text{GSV}}{}^{-\frac{1}{2}} \overrightarrow{A}_c, \mathbf{Q}_{\text{GSV}}{}^{-\frac{1}{2}} \overrightarrow{A}_s \right\rangle$	9.32%	–	0.0363	–
	Linearized LR	$K_{\text{LR}}\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) = \left\langle \boldsymbol{Q}_{\text{LR}}{}^{-\frac{1}{2}} \overrightarrow{A}_c, \boldsymbol{Q}_{\text{LR}}{}^{-\frac{1}{2}} \overrightarrow{A}_s \right\rangle$	–	22.69%	–	0.0865
	GLDS	$K_{\text{GLDS}}\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) = \left\langle \boldsymbol{Q}_{\text{GLDS}}{}^{-\frac{1}{2}} \overrightarrow{A}_c, \boldsymbol{Q}_{\text{GLDS}}{}^{-\frac{1}{2}} \overrightarrow{A}_s \right\rangle$	14.56%	25.67%	0.0647	0.0928
	Regression	$K_{\text{Reg}}\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) = k(\overrightarrow{A}_c, \cdot)^{\text{T}}(\mathbf{K\Gamma K} + \lambda \mathbf{K})^{-1} k(\overrightarrow{A}_s, \cdot)$	**8.84%**	22.19%	**0.0335**	0.0857
	Max-Margin	$K_{\text{MM-Emp}}\left(\overrightarrow{A_c}, \overrightarrow{A_s}\right) = < \mathbf{F}_b^{-\frac{1}{2}} \mathbf{f}_c, \mathbf{F}_b^{-\frac{1}{2}} \mathbf{f}_s >$	9.14%	**21.68%**	0.0332	**0.0811**
LR	—	$S_{\text{LR}} = \frac{1}{T} \sum_{t=1}^{T} \log p_s(\boldsymbol{o}_t)/p_b(\boldsymbol{o}_t)$	*10.29%*	*23.79%*	*0.0428*	*0.0916*
Kernel + LR	GSV	$\eta \frac{S_{\text{Ker}}(\boldsymbol{O}) - \mu_{\text{Ker}}}{\sigma_{\text{Ker}}} + (1-\eta) \frac{S_{\text{LR}}(\boldsymbol{O}) - \mu_{\text{LR}}}{\sigma_{\text{LR}}}$	8.51%	–	0.0351	–
	Regression		8.21%	–	0.0318	–
	Linearized LR		–	22.17%	–	0.0849
	Max-Margin		–	21.38%	–	0.0809
	All		***7.74%***	***21.23%***	***0.0281***	***0.0807***
High + Low	All	$\eta \frac{S_{\text{High}}(\boldsymbol{O}) - \mu_{\text{High}}}{\sigma_{\text{High}}} + (1-\eta) \frac{S_{\text{Low}}(\boldsymbol{O}) - \mu_{\text{Low}}}{\sigma_{\text{Low}}}$	***7.32%***		***0.0276***	

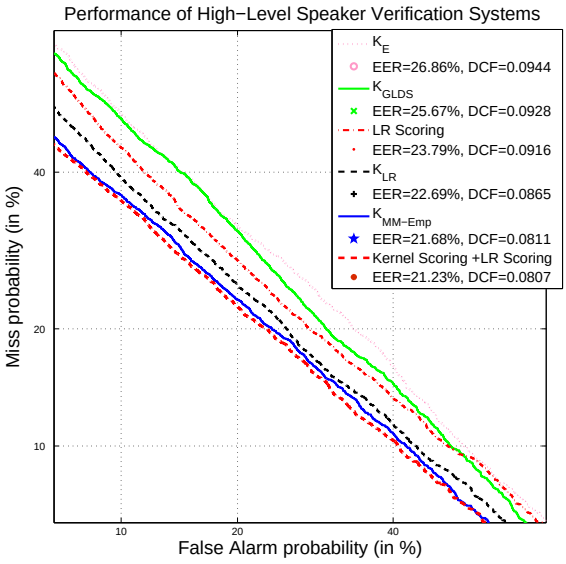

Fig. 4. DET performance of high-level systems using different kernels scoring approaches and the fusion of kernel scoring and likelihood-ratio (LR) scoring. The legends are arranged in decreasing EER.

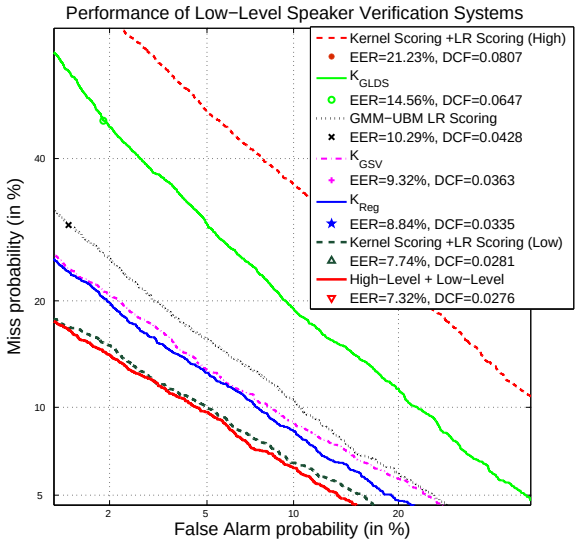

Fig. 5. DET performance of low-level systems using different kernel scoring approaches and the fusion of all high-level systems and all low-level systems. The legends are arranged in decreasing EER.

For high-level speaker verification, Fig. 4 shows that the performance of linearized LR kernel scoring (K_{LR}) is significantly better than that of LR scoring (red dashed-dot) across a wide range of decision thresholds. This is mainly attributed to the explicit use of discriminative information in the kernel function of the SVM and to the optimal selection of background speakers by SVM training. Although LR scoring also considers the impostor information, it can only implicity use this information through the UBM. In LR kernel scoring, on the other hand, the SVM of each target speaker

is discriminatively trained to differentiate the target speaker from all of the background speakers. The SVM effectively provides an optimal set of weights for this differentiation. On the other hand, in LR scoring, all target speakers share the same background model and the weight is always identical ($= -1$) across all target speakers. This explains the superiority of the kernel scoring approach.

B. Effect of the Kernel Normalization

Among all the kernels, only the Euclidean kernel (K_{E}) does not use normalization, i.e., does not pre-multiply the supervectors by a normalization matrix during kernel evaluation. Comparing the EER of K_{E} and other kernels in Table IV suggests that normalization can help improve performance, which is consistent with the results of [18]. The reason is that normalizing the supervectors by the background models can prevent some features (with large numerical values) from dominating the SVM scoring.

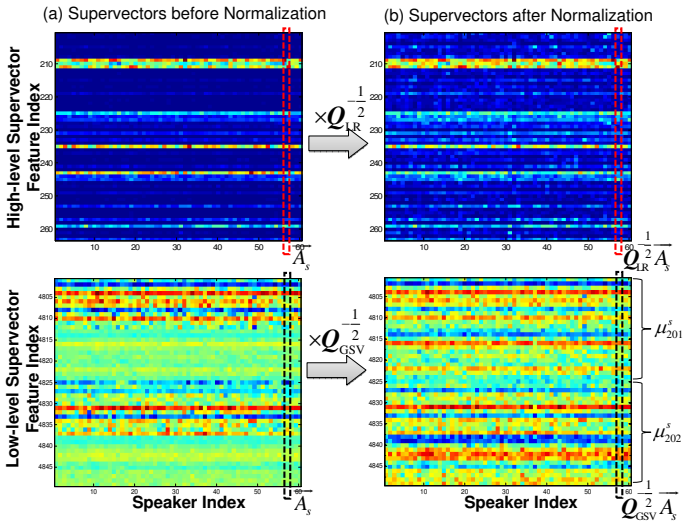

Fig. 6. The effect of the normalization term $\boldsymbol{Q}^{-\frac{1}{2}}$ (see Eq. 8 and 5). Each column in the images represents the supervector of one target speaker. Upper panel: High-level system. Lower panel: Low-level system.

To highlight the importance of normalization, let us consider the matrix $\boldsymbol{Q}_{\mathrm{GSV}}$ (Eq. 5) in the GSV kernel and $\boldsymbol{Q}_{\mathrm{LR}}$ (Eq. 8) in the linearized LR kernel. Figures 6(a) and 6(b) display the un-normalized $\overrightarrow{A_s}$ and the normalized $\overrightarrow{A_s}$ for 150 speakers, respectively. Evidently, without normalization (Fig. 6(a)), some features have a large but almost constant value across all target speakers, e.g., rows with dark-blue in Fig. 6(a). These features will cause problems in SVM classification because they will dictate the decision boundary of the SVM, even though they contain little speaker-dependent information. This problem has been largely alleviated by the normalization, as demonstrated in Fig. 6(b). In particular, the normalization has the effect of keeping all features within a comparable range, which helps to prevent the large but almost constant features from dominating the classification decision.

C. Compare Different Kernel Scoring

Table IV suggests that the proposed regression optimized kernel and maximum-margin empirical kernel are the best among all kernels that we evaluated. The differences in EER between GSV and regression kernels and between GSV and max-margin kernels have p-values [39] smaller than 0.01. This suggests that optimizing a general discriminant function (Eq. 11 and Eq. 20) to derive a kernel is better than (a) using a specific distance metric (e.g., Euclidean kernel K_{E} and GSV kernel K_{GSV}) and (b) assigning a specific form for the discriminant function as in the linearized LR kernel (Eq. 7) and the GLDS kernel.

D. Fusions of Kernel Scoring and LR Scoring

Figures 4 and 5 demonstrate that the fusion of LR scores and the SVM scores leads to better performance across a wide range of decision thresholds for both high-level (red-dashed in Fig. 4) and low-level (dark-green dashed in Fig. 5) cases.

E. Fusions of High- and Low-Level Systems

Table IV and Fig. 5 (solid red) show that the performance can be further improved by fusing the high-level systems and the low-level systems, resulting in an EER of 7.32%, with a p-value [39] smaller than 0.0001 when compared with the EER (7.74%) without fusion. This EER is also lower than other recent results (e.g., [40]) on the same corpus in the literature.

VIII. Conclusions and Future Work

This paper provides theoretical and experimental evidences to demonstrate that kernel-based SVM scoring is superior to frame-based LR scoring in speaker verification. The paper proposes an optimization procedure for kernel construction, which results in two discriminative kernels that are more general than the existing ones. Results show that the proposed optimized regression kernel and maximum-margin empirical kernel outperform the GSV kernel and LR scoring. This suggests that optimizing a general discriminant function to derive a kernel is better than (a) using a specific distance metric (e.g., GSV kernel) and (b) assigning a specific form for the discriminant function as in the linearized LR kernel and the GLDS kernel. Results also show that the fusion of LR scoring and kernel scoring can further reduce the EER in both high- and low-level speaker verification. The performance can be further improved by linearly fusing the high- and low-level systems, resulting in an EER of 7.32%. Although the proposed kernels are evaluated on a speaker verification task, they are general enough for other classification problems.

We notice that the dimensionality of the supervectors is fairly high (6144 for low-level systems and 720 for high-level systems) and that many of the dimensions have low variances [41]. This suggests that some of the dimensions could be discarded. Another possibility is to find the optimal discriminant subspace and project the supervectors into the subspace. As the number of training vectors for each target speaker is significantly smaller than the feature dimension, the discriminative common vector method [42] could be a potential candidate for this purpose.

APPENDIX

This appendix derives Eq. 2 for continuous generative models in Table I. Specifically, it shows that the LR score of a GMM-UBM system can be expressed as a function of supervectors derived from the claimant, target-speaker, and UBM. For the case of discrete models, see [17].

Assume that the continuous model for a speaker can be expressed as a Gaussian mixture density function: $p(\boldsymbol{o}) = \sum_{i=1}^{G} \lambda_i \mathcal{N}(\boldsymbol{o}\ ; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where λ_i are mixture weights, $\mathcal{N}(\cdot)$ represents a Gaussian density with mean vector $\boldsymbol{\mu}_i$, and covariance matrix $\boldsymbol{\Sigma}_i$. Given a GMM, a supervector $\overrightarrow{\mu}$ can be obtained by stacking all of the GMM mean vectors [18].

Given an utterance $\boldsymbol{O}_c = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ from claimant c, the LR score of the utterance can be obtained by averaging the individual log-likelihood ratios as follows:[6]

$$S_{\text{LR}}(\boldsymbol{O}_c, s) = \frac{1}{T} \sum_{t=1}^{T} \log \frac{\sum_{i=1}^{G} \lambda_i^s \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s)}{\sum_{i=1}^{G} \lambda_i^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)}, \tag{27}$$

where s and b represent the target speaker and background speakers, respectively, and T is the number of frames in the utterance.

The following Lemma will be used for deriving the supervector-based formulation.

Lemma 1 *Let $a_i, b_i \geq 0$ for $i = 1, \ldots, n$, then*

$$\log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \geq \frac{1}{\sum_{j=1}^{n} b_j} \sum_{i=1}^{n} \left(b_i \log \frac{a_i}{b_i} \right), \tag{28}$$

with equality iff $\frac{a_i}{b_i}$ = constant. This lemma can be proved using Jensen inequality.

Lemma 2 *Let $a_i, b_i \geq 0$ for $i = 1, \ldots, n$, then*

$$\log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \leq \frac{1}{\sum_{j=1}^{n} a_j} \sum_{i=1}^{n} \left(a_i \log \frac{a_i}{b_i} \right). \tag{29}$$

This lemma is the log-sum inequality used in [18], [43].

Let $a_i = \lambda_i^s \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s)$ and $b_i = \lambda_i^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)$. Using Lemma 2 and Lamma 1, we can write:

$$\begin{aligned}
&\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{G} \frac{\lambda_i^s \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s)}{\sum_{j=1}^{G} \lambda_j^s \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_j^s, \boldsymbol{\Sigma}_j^s)} \log \frac{\lambda_i^s \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s)}{\lambda_i^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)} \geq \\
&S_{\text{LR}}(\boldsymbol{O}_c, s) = \frac{1}{T} \sum_{t=1}^{T} \log \frac{\sum_{i=1}^{G} \lambda_i^s \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s)}{\sum_{i=1}^{G} \lambda_i^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)} \\
&\geq \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{G} \frac{\lambda_i^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)}{\sum_{j=1}^{G} \lambda_j^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_j^b, \boldsymbol{\Sigma}_j^b)} \log \frac{\lambda_i^s \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^s)}{\lambda_i^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)}.
\end{aligned} \tag{30}$$

Denote $\gamma_i^b(t) = \frac{\lambda_i^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)}{\sum_{j=1}^{G} \lambda_j^b \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_j^b, \boldsymbol{\Sigma}_j^b)}$ as the posterior probability that $\boldsymbol{o}_t$ is generated by the i-th mixture in the UBM. (Similarly, $\gamma_i^s(t)$ for speaker s.) In GMM-UBM systems, the MAP algorithm [2] is applied to the mean vectors only. Therefore, $\boldsymbol{\Sigma}_i^s = \boldsymbol{\Sigma}_i^b$ and $\lambda_i^s = \lambda_i^b$. As a result, Eq. 30 can be expressed as:

$$\begin{aligned}
&\frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{G} \gamma_i^s(t) \log \frac{\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^b)}{\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)} \geq S_{\text{LR}}(\boldsymbol{O}_c, s) \\
&\geq \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{G} \gamma_i^b(t) \log \frac{\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^b)}{\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)}.
\end{aligned} \tag{31}$$

Because every speaker model is adapted from the same UBM, the difference between the upper bound and the lower bound of the LR score in Eq. 31 is finite. Therefore, we can use the lower bound as an approximation to the LR score:

$$\begin{aligned}
S_{\text{LR}}(\boldsymbol{O}_c, s) &\doteq \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{G} \gamma_i^b(t) \log \frac{\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^s, \boldsymbol{\Sigma}_i^b)}{\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)} \\
&= \frac{1}{2T} \sum_{i=1}^{G} \sum_{t=1}^{T} \gamma_i^b(t) \Big\{ -(\boldsymbol{o}_t - \boldsymbol{\mu}_i^s)^{\text{T}} (\boldsymbol{\Sigma}_i^b)^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu}_i^s) \\
&\qquad + (\boldsymbol{o}_t - \boldsymbol{\mu}_i^b)^{\text{T}} (\boldsymbol{\Sigma}_i^b)^{-1} (\boldsymbol{o}_t - \boldsymbol{\mu}_i^b) \Big\} \\
&= \frac{1}{2T} \sum_{i=1}^{G} \sum_{t=1}^{T} \gamma_i^b(t) \\
&\qquad \Big\{ \left[(\boldsymbol{o}_t - \boldsymbol{\mu}_i^s) + (\boldsymbol{o}_t - \boldsymbol{\mu}_i^b) \right]^{\text{T}} (\boldsymbol{\Sigma}_i^b)^{-1} (\boldsymbol{\mu}_i^s - \boldsymbol{\mu}_i^b) \Big\}.
\end{aligned} \tag{32}$$

For $\boldsymbol{o}_t$ belonging to claimant c, we express the MAP-adapted mean of the i-th Gaussian at iteration k as:

$$\boldsymbol{\mu}_i^{c,(k)} = \frac{\sum_{t=1}^{T} \gamma_i^{c,(k-1)}(t) \boldsymbol{o}_t + \tau \boldsymbol{\mu}_i^b}{\sum_{t=1}^{T} \gamma_i^{c,(k-1)}(t) + \tau}, \tag{33}$$

where $\gamma_i^{c,(0)}(t) \triangleq \gamma_i^b(t)$ and τ is the MAP adaptation relevance factor which controls the influence of the prior distribution on the final model. Assuming that $T \gg \tau$ and that only one iteration is performed, we have:

$$\boldsymbol{\mu}_i^c \doteq \frac{\sum_{t=1}^{T} \gamma_i^b(t) \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_i^b(t)}. \tag{34}$$

The maximum-likelihood estimates of the i-th mixture weight in the UBM is given by:

$$\lambda_i^{b,(k)} = \frac{1}{T} \sum_{t=1}^{T} \gamma_i^{b,(k-1)}(t). \tag{35}$$

Substituting Eq. 35 into Eq. 34, we have:

$$\sum_{t=1}^{T} \gamma_i^b(t) \boldsymbol{o}_t = T \lambda_i^b \boldsymbol{\mu}_i^c. \tag{36}$$

[6] Note that λ_i^b here is equivalent to $\lambda_{b,i}$ in Table I.

Substituting Eq. 36 into Eq. 32, we have:

$$
\begin{aligned}
S_{\text{LR}}(\boldsymbol{O}_c, s) \doteq & \frac{1}{2}\sum_{i=1}^{G}\Big\{\left[(\lambda_i^b\boldsymbol{\mu}_i^c - \lambda_i^b\boldsymbol{\mu}_i^s) + (\lambda_i^b\boldsymbol{\mu}_i^c - \lambda_i^b\boldsymbol{\mu}_i^b)\right]^{\text{T}} \\
& (\boldsymbol{\Sigma}_i^b)^{-1}(\boldsymbol{\mu}_i^s - \boldsymbol{\mu}_i^b)\Big\} \\
= & \sum_{i=1}^{G}\left(\sqrt{\lambda_i^b}(\boldsymbol{\Sigma}_i^b)^{-\frac{1}{2}}\boldsymbol{\mu}_i^c\right)^{\text{T}}\left(\sqrt{\lambda_i^b}(\boldsymbol{\Sigma}_i^b)^{-\frac{1}{2}}\boldsymbol{\mu}_i^s\right) - \\
& \sum_{i=1}^{G}\left(\sqrt{\lambda_i^b}(\boldsymbol{\Sigma}_i^b)^{-\frac{1}{2}}\boldsymbol{\mu}_i^c\right)^{\text{T}}\left(\sqrt{\lambda_i^b}(\boldsymbol{\Sigma}_i^b)^{-\frac{1}{2}}\boldsymbol{\mu}_i^b\right) + d_s \\
= & \overrightarrow{A_c}^{\text{T}}\boldsymbol{Q}_{\text{GSV}}^{-1}\overrightarrow{A_s} - \overrightarrow{A_c}^{\text{T}}\boldsymbol{Q}_{\text{GSV}}^{-1}\overrightarrow{A_b} + d_s,
\end{aligned}
\tag{37}
$$

where $\boldsymbol{Q}_{\text{GSV}} = diag\left\{(\lambda_1^b)^{-1}diag\left(\boldsymbol{\Sigma}_1^b\right), \ldots, (\lambda_G^b)^{-1}diag\left(\boldsymbol{\Sigma}_G^b\right)\right\}$, $\overrightarrow{A} = [\boldsymbol{\mu}_1^{\text{T}}, \ldots, \boldsymbol{\mu}_G^{\text{T}}]^{\text{T}}$ is a supervector formed by stacking all of the Gaussians mean vectors, and

$$
d_s = -\frac{1}{2}\overrightarrow{A_s}^{\text{T}}\boldsymbol{Q}_{\text{GSV}}^{-1}\overrightarrow{A_s} + \frac{1}{2}\overrightarrow{A_b}^{\text{T}}\boldsymbol{Q}_{\text{GSV}}^{-1}\overrightarrow{A_b}.
$$

Note that $\overrightarrow{A_c}^{\text{T}}\boldsymbol{Q}_{\text{GSV}}^{-1}\overrightarrow{A_s}$ in Eq. 37 is actually the GMM-supervector (GSV) kernel in [18] and that unlike [18] the above derivation derives the GSV kernel from the likelihood ratio in Eq. 27.

REFERENCES

[1] S. X. Zhang and M. W. Mak, "Optimization of discriminative kernels in SVM speaker verification," in *Interspeech'09*, Brighton, Sept 2009, pp. 1275–1278.

[2] 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.

[3] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on ASSP*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[4] D. A. Reynolds, "Channel robust speaker verification via feature mapping," in *Proc. ICASSP*, Hong Kong, China, Apr. 2003, vol. 2, pp. 53–56.

[5] M. W. Mak, K. K. Yiu, and S. Y. Kung, "Probabilistic feature-based transformation for speaker verification over telephone networks," *Neurocomputing, Special Issue on Neural Networks for Speech and Audio Processing*, vol. 71, no. 1–3, pp. 137–146, Dec. 2007.

[6] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, Apr. 1995.

[7] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 42–54, Jan. 2000.

[8] A. Solomonoff, C. Quillen, and W. M. Campbell, "Channel compensation for SVM speaker recognition," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, Toledo, Spain, May–June 2004, pp. 57–62.

[9] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. ICASSP*, Toulouse, France, May 2006, vol. 1, pp. 97–100.

[10] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[11] D. Klusacek, J. Navratil, D. A. Reynolds, and J. P. Campbell, "Conditional pronunciation modeling in speaker detection," in *Proc. ICASSP*, Hong Kong, China, Apr. 2003, vol. 4, pp. 804–807.

[12] D. Reynolds, et. al., "The superSID project: Exploiting high-level information for high-accuracy speaker recognition," in *Proc. ICASSP*, Hong Kong, China, Apr. 2003, vol. 4, pp. 784–787.

[13] E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke, "Modeling prosodic feature sequences for speaker recognition," *Speech Communication*, vol. 46, no. 3–4, pp. 455–472, Jul. 2005.

[14] S. X. Zhang, M. W. Mak, and Helen H. Meng, "Speaker verification via high-level feature based phonetic-class pronunciation modeling," *IEEE Trans. on Computers*, vol. 56, no. 9, pp. 1189–1198, Aug. 2007.

[15] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proc. ICASSP*, Orlando, USA, May 2002, vol. 1, pp. 161–164.

[16] W. M. Campbell, J. R. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "High-level speaker verification with support vector machines," in *Proc. ICASSP*, Montreal, Canada, May 2004, vol. 1, pp. 73–76.

[17] S. X. Zhang and M. W. Mak, "High-level speaker verification via articulatory-feature based sequence kernels and SVM," in *Proc. Interspeech*, Brisbane, Australia, Sept. 2008, pp. 1393–1396.

[18] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, May 2006.

[19] V. Wan and S. Renals, "SVMSVM: Support vector machine speaker verification methodology," in *Proc. ICASSP'03*, Hong Kong, China, Apr. 2003, vol. 2, pp. 221–224.

[20] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[21] B. Schölkopf, J. Weston, E. Eskin, C. Leslie, and W. S. Noble, "A kernel approach for learning from almost orthogonal patterns," in *Proc. 13th European Conference on Machine Learning*, Helsinki, Finland, Aug. 2002, pp. 511–528.

[22] K. Tsuda., "Support vector classifier with asymmetric kernel functions," in *Proc. ESANN*, Bruges, Belgium, Apr. 1999, pp. 183–188.

[23] 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.

[24] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Trans. of the London Philosophical Society (A)*, vol. 209, pp. 415–446, 1909.

[25] Q. Jin, J. Navratil, D.A. Reynolds, J.P. Campbell, W.D. Andrews, and J.S. Abramson, "Combining cross-stream and time dimensions in phonetic speaker recognition," in *Proc. ICASSP*, Hong Kong, China, Apr. 2003, vol. 4, pp. 800–803.

[26] A. E. Rosenberg, J. DeLong, C. H. Lee, B. H. Juang, and F. K. Soong, "The use of cohort normalized scores for speaker verification," in *Proc. ICSLP'92*, 1992, vol. 2, pp. 599–602.

[27] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambrigde University Press, Cambridge, UK, 2004.

[28] F. Girosi, "An equivalence between sparse approximation and support vector machines," *Neural Computation*, vol. 10, no. 6, pp. 1455–1480, Aug. 1998.

[29] M. Hu, Y. Chen, and J. T. Y. Kwok, "Building sparse multiple-kernel SVM classifiers," *IEEE Transactions on Neural Networks*, vol. 20, no. 5, pp. 827–839, 2009.

[30] 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.

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

[32] "The NIST year 2001 speaker recognition evaluation plan," *http://www.nist.gov/speech/tests/sre/2001/index.html*.

[33] K. Y. Leung, M. W. Mak, M. H. Siu, and S. Y. Kung, "Adaptive articulatory feature-based conditional pronunciation modeling for speaker verification," *Speech Communication*, vol. 48, no. 1, pp. 71–84, Jan. 2006.

[34] S. X. Zhang and M. W. Mak, "A new adaptation approach to high-level speaker-model creation in speaker verification," *Speech Communication*, vol. 51, no. 6, pp. 534–550, Jun. 2009.

[35] "The NIST year 2002 speaker recognition evaluation plan," *http://www.nist.gov/speech/tests/sre/2002/index.html*.

[36] J. P. Campbell and D. A. Reynolds, "Corpora for the evaluation of speaker recognition systems," in *Proc. ICASSP*, Phoenix, USA, Mar. 1999, vol. 2, pp. 829–832.

[37] D. A. Reynolds, "HTIMIT and LLHDB: Speech corpora for the study of handset transducer effects," in *Proc. ICASSP*, Munich, Germany, Apr. 1997, vol. 2, pp. 1535–1538.

[38] "SVMlight," *http://svmlight.joachims.org/*.

[39] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, Glasgow, UK, May 1989, vol. 1, pp. 532–535.

[40] C. Longworth and M. J. F. Gales, "Combining derivative and parametric kernels for speaker verification," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 17, no. 4, pp. 745–757, May 2009.

[41] M. W. Mak and W. Rao, "Utterance Partitioning with Acoustic Vector Resampling for GMM–SVM Speaker Verification," *Speech Communication*, 2010, in press.

[42] H. Cevikalp, M. Neamtu, and M. Wilkes, "Discriminative common vector method with kernels," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1550–1565, 2006.

[43] M. N. Do, "Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models," *IEEE Signal Processing Letters*, vol. 10, no. 4, pp. 115–118, Apr. 2003.