Pairwise Discriminative Speaker Verification in the I-Vector Space

Pairwise Discriminative Speaker Verification in the I-Vector Space

Sandro Cumani, Niko Brümmer, Lukáš Burget, Pietro Laface,
Oldřich Plchot and Vasileios Vasilakakis

Abstract

This work presents a new and efficient approach to discriminative speaker verification in the i–vector space. We illustrate the development of a linear discriminative classifier that is trained to discriminate between the hypothesis that a pair of feature vectors in a trial belong to the same speaker or to different speakers. This approach is alternative to the usual discriminative setup that discriminates between a speaker and all the other speakers. We use a discriminative classifier based on a Support Vector Machine (SVM) that is trained to estimate the parameters of a symmetric quadratic function approximating a log–likelihood ratio score without explicit modeling of the i–vector distributions as in the generative Probabilistic Linear Discriminant Analysis (PLDA) models. Training these models is feasible because it is not necessary to expand the i–vector pairs, which would be expensive or even impossible even for medium sized training sets. The results of experiments performed on the tel-tel extended core condition of the NIST 2010 Speaker Recognition Evaluation are competitive with the ones obtained by generative models, in terms of normalized Detection Cost Function and Equal Error Rate. Moreover, we show that it is possible to train a gender–independent discriminative model that achieves state–of–the–art accuracy, comparable to the one of a gender–dependent system, saving memory and execution time both in training and in testing.

Index Terms

Speaker Recognition, I-vector, Discriminative training, Probabilistic Linear Discriminant Analysis, Support Vector Machines, Large–scale training.

I. Introduction

RECENT developments in speaker recognition technology have seen the success of systems based on a low–dimensional representation of a speech segment, the so–called “identity vector” or i–vector [1], [2]. An i–vector is a compact representation of a Gaussian Mixture Model (GMM) supervector [3], which captures most of the GMM supervectors variability. The availability of low–dimensional features boosted the research interest towards probabilistic generative models [4]. These techniques aim at decomposing the speaker and inter–session variability components of i–vectors, estimating their distributions, and perform induction on the speaker identity in a Bayesian framework. The most effective approaches in this framework are the Gaussian (G–PLDA) or Heavy-Tailed Probabilistic Linear Discriminant Analysis (HT–PLDA) [4], and the Two-covariance model, a linear-Gaussian generative model introduced in [5], [6]. PLDA models [7] not only have well founded probabilistic interpretations, but have also the advantage of producing log–likelihood ratios which do not, in principle, require score normalization. In [4] this has been confirmed in the case of telephone speech, for heavy-tailed distributions, whereas normalization was needed for Gaussian distributions. A complete symmetry of the train and test segments is another interesting characteristic of these approaches.

Besides generative models, remarkable success has been also obtained by discriminative systems based on Support Vector Machines, usually in combination with Nuisance Attribute Projection [8], [9] for inter–session compensation.

Sandro Cumani, Pietro Laface and Vasileios Vasilakakis are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, 10143 Torino, Italy (e-mail: sandro.cumani@polito.it, pietro.laface@polito.it, vasileios.vasilakakis@polito.it).

Sandro Cumani is currently also with Brno University of Technology, supported by Czech Ministry of Education project No. CZ.1.07/2.3.00/30.0005.

Niko Brümmer is with AGNITIO Research, Stellenbosch, South Africa, (e-mail: niko.brummer@gmail.com).

Lukáš Burget and Oldřich Plchot are with Brno University of Technology, Czech Republic, (e-mail:burget@fit.vutbr.cz, iplchot@fit.vutbr.cz). They were partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20015, by Czech Ministry of Education project No. MSM0021630528 and by IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

Vasileios Vasilakakis is supported by the FP7/2007-2013 European Programme under grant agreement n.238803

However, SVM–based systems have mostly been trained as one–versus–all classifiers, i.e., using the utterances of a given speaker against the utterances from a background cohort of impostor speakers. This approach has a major weakness: the available samples for the target speaker are often scarce, and can easily reduce to just one. Moreover, in a scenario where a single enrollment and test utterance are available for a speaker, the two utterances play a completely different role, which implies that the score for a given trial is not symmetric with respect to the segments.

In this work we present a new framework for discriminative speaker classification that aims at overcoming the problems of the classical SVM approach while retaining most of the interesting characteristics of Bayesian systems, namely almost calibrated scores and symmetry between enrollment and test utterances. In this approach we do not model speaker classes, but we train a binary classifier which classifies a pair of utterances as belonging to either the *same speaker* or *different speakers* [1]. In particular, the speaker verification score for a pair of i-vectors is computed using a function having a form derived from the PLDA generative model. The parameters of the function, however, are estimated using a discriminative training criterion. Discriminative training of a PLDA-like model for speaker verification was originally proposed in [5], and some preliminary work was done in [1] using as features the speaker factors extracted using Joint Factor Analysis [10].

We show that the same functional form derived from PLDA can be obtained without making reference to the distribution of the i–vectors, and that we can train an SVM that estimates the parameters of a second order approximation of good symmetric score functions using an expansion of each i–vector pair. We also show that this pairwise SVM corresponds to a second degree polynomial kernel SVM.

Experiments performed on a NIST SRE 2010 evaluation task [11] show that this new approach achieves state–of–the–art performance with a scoring time comparable to the simplest i–vector based systems. Moreover, our approach was directly used to train a gender–independent speaker recognition system, ignoring the gender labels both in training and in test, with accuracy comparable to the one of gender–dependent systems trained on the same data.

The outline of the paper is as follows: Section II briefly introduces the i–vectors, and Section III recalls the PLDA approach and the two–covariance model, where both the speaker and the intra–speaker variability sub–spaces are assumed to be full–rank. It also shows how to obtain a binary linear classifier in an appropriate nonlinearly expanded space of i-vector pairs. In Section IV, using an expanded vector representing a pair of i–vectors in a trial, we derive an SVM model. A fast solution to the computation of gradient and score, which are needed for efficient training and scoring, is presented in Section VI. The experimental results comparing the performance of the discriminative and generative models are given in Section VII, and conclusions are drawn in Section VIII.

II. I–VECTORS

I–vector based techniques represent the state–of–the–art in speaker verification [2], [12]. I–vectors provide an elegant way of reducing large-dimensional input data. In this approach, a speech segment is mapped to a fixed small-dimensional vector retaining most of the relevant information necessary to give state-of-the-art speaker recognition performance. The mapping is obtained by modeling the sequence of feature vectors by a large GMM, the parameters of which are constrained to lie in a low dimensional subspace. In particular, the i–vector model constrains the GMM supervector $\mathbf{s}$, representing both the speaker and inter–session characteristics of a given speech segment, to live in a single subspace according to:

$$\mathbf{s} = \mathbf{m} + \mathbf{T}\boldsymbol{\phi} \ , \tag{1}$$

where $\mathbf{m}$ is the Universal Background Model (UBM) GMM mean supervector, with C GMM components of dimension F. $\mathbf{T}$ is a low-rank rectangular matrix, of $C \times F$ rows and M columns, spanning the subspace including important inter– and intra–speaker variability in the mean supervector space, and $\boldsymbol{\phi}$ is a realization of a latent variable $\boldsymbol{\Phi}$ of size M with standard normal distribution. A Maximum-Likelihood estimate of matrix T is usually obtained by minor modifications of the Joint Factor Analysis approach [10]. Given the sequence of features representing an utterance, $\mathcal{X}$, its i-vector is computed as the Maximum a Posteriori (MAP) point estimate of the variable $\boldsymbol{\Phi}$, i.e., the mean of the posterior distribution $p(\boldsymbol{\Phi}|\mathcal{X})$.

The main advantage of the i–vector representation is that the problem of intersession variability can be deferred to a second stage. The possibility of dealing in this second stage with low-dimensional vectors, rather than with the high-dimensional supervectors of the GMM means, boosted the study of probabilistic generative models [4], [6]. A procedure for extracting i-vectors has been described and effectively used in [2], [12].

III. GENERATIVE MODELS

Good speaker recognition accuracy has been obtained using i–vectors and simple LDA and cosine distance scoring [2]. However, since the introduction of these low–dimensional features, the speaker recognition community has focused on more accurate models for computing speaker detection scores directly from i–vectors. The generative models analyzed in [7], [4] are among the best models for comparison of i–vectors. In this section we briefly recall the PLDA framework and a simplified model that will be used for deriving the formulation of our discriminative speaker verification approach.

A. PLDA

Probabilistic Linear Discriminant Analysis (PLDA) [7], [4] is one of the most successful models for i–vectors comparison. PLDA assumes that the i–vector generation process can be described by means of a latent variable probabilistic model where i–vector $\boldsymbol{\phi}$ is modeled as the sum of three factors, namely a speaker factor $\mathbf{y}$, an inter–session (channel) factor $\mathbf{x}$ and the residual noise ϵ as:

$$\boldsymbol{\phi} = \mathbf{m} + \mathbf{U}_1\mathbf{y} + \mathbf{U}_2\mathbf{x} + \epsilon \ . \tag{2}$$

Matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ typically constrain the speaker and inter–session factors to be of lower dimension than the i–vectors space. The generation of an i-vector requires choosing a random speaker factor $\mathbf{y}$ according to speaker prior distribution $p(\mathbf{y})$ and a random inter–session factor $\mathbf{x}$ according to a prior distribution $p(\mathbf{x})$. The i-vector is then the sum of $\mathbf{U}_1\mathbf{y} + \mathbf{U}_2\mathbf{x}$, the mean vector $\mathbf{m}$ and of the residual noise $\boldsymbol{\epsilon}$ generated according to the distribution $p(\boldsymbol{\epsilon})$.

PLDA estimates the matrices $\mathbf{U}_1$, $\mathbf{U}_2$, and the values of the hyper–parameters of possible parametric priors [4], which maximize the likelihood of the observed i–vectors, assuming that i–vectors from the same speaker share the same speaker factor, i.e., the same value for latent variable $\mathbf{y}$.

The simplest PLDA model (G-PLDA) assumes a Gaussian distribution for the prior parameters. However, in [4] it is shown that ML estimation of the PLDA parameters under a Gaussian assumption fails to produce accurate models for i-vectors. Thus, heavy–tailed distributions for the model priors have been proposed leading to the Heavy-Tailed PLDA model, which however, is computationally expensive.

A simpler approach preserves the Gaussian distribution assumption, but incorporates a pre–processing step where the vector dimensionality is possibly further reduced by LDA, and more importantly, within-class covariance and length normalization is applied to the resulting patterns [13]. Using these dimension reduced and normalized i–vectors, the performance of the Heavy–Tailed and Gaussian PLDA models is comparable, the latter being much faster both in training and in testing.

B. Two-covariance model

Further model simplification is obtained by merging together the residual noise and the inter–session components, assuming that the speaker and inter–session subspaces span the entire i–vector subspace. This simplified model is referred to as the two–covariance model [5], [6]. An i-vector $\boldsymbol{\phi}$ is assumed to be produced by a linear-Gaussian generative model $\mathcal{M}$ that accounts for a speaker $\mathbf{y}$ and a Gaussian–distributed component $\mathbf{z}$, including inter–session variability, as:

$$\boldsymbol{\phi} = \mathbf{y} + \mathbf{z} \ . \tag{3}$$

If we assume that the speaker component is Gaussian–distributed as:

$$P(\mathbf{y}|\mathcal{M}) = \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \mathbf{B}^{-1}) \ , \tag{4}$$

where $\mathbf{B}^{-1}$ is the between–speaker covariance matrix, and the distribution of the i–vector given the speaker identity is also Gaussian:

$$P(\boldsymbol{\phi}|\mathbf{y}, \mathcal{M}) = \mathcal{N}(\boldsymbol{\phi}|\mathbf{y}, \mathbf{W}^{-1}) \ , \tag{5}$$

where $\mathbf{W}^{-1}$ is the within–speaker covariance matrix, then, given a set $\mathcal{S} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n\}$ of n i-vectors associated to the same speaker, the posterior of y is also normal [14]:

$$P(\mathbf{y}|\mathcal{S}, \mathcal{M}) = \mathcal{N}(\mathbf{y}|\mathbf{L}^{-1}\boldsymbol{\gamma}, \mathbf{L}^{-1}) \ , \tag{6}$$

and the parameters of the distribution are:

$$\mathbf{L} = \mathbf{B} + n\mathbf{W} \qquad\qquad \boldsymbol{\gamma} = \mathbf{B}\boldsymbol{\mu} + \mathbf{W}\sum_{\boldsymbol{\phi}\in S}\boldsymbol{\phi}\ . \tag{7}$$

C. Two–covariance scoring

The conditional likelihood of two i-vectors allows obtaining the speaker verification log–likelihood ratio score between the "same–speaker" hypothesis H_s and "different–speaker" hypothesis H_d:

$$\lambda = \log \frac{P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|H_s)}{P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|H_d)}\ , \tag{8}$$

where $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2$ are two i-vectors that are scored.
The numerator probability is computed assuming that the i–vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ belong to the same speaker, i.e they share a common value of the hidden variable $\mathbf{y}$. According to Bayes rule this probability can be computed as:

$$P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|H_s) = \frac{P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathbf{y}_0, \mathcal{M})P(\mathbf{y}_0|\mathcal{M})}{P(\mathbf{y}_0|\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)}\ , \tag{9}$$

where $\mathbf{y}_0$ is any value which does not cause the denominator to be zero. Since the intersession variability components of different utterances are assumed to be independent, i.e., the i–vectors are independent given the speaker variable, (9) can be rewritten as:

$$P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|H_s) = \frac{P(\boldsymbol{\phi}_1|\mathbf{y}_0, \mathcal{M})P(\boldsymbol{\phi}_2|\mathbf{y}_0, \mathcal{M})P(\mathbf{y}_0|\mathcal{M})}{P(\mathbf{y}_0|\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)}\ . \tag{10}$$

The denominator probability in (8) is computed, instead, assuming that the i–vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ belong to different speakers, as:

$$\begin{gathered} P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|H_d) = P(\boldsymbol{\phi}_1)P(\boldsymbol{\phi}_2) = \\ \frac{P(\boldsymbol{\phi}_1|\mathbf{y}_0, \mathcal{M})P(\mathbf{y}_0|\mathcal{M})}{P(\mathbf{y}_0|\boldsymbol{\phi}_1, \mathcal{M})} \cdot \frac{P(\boldsymbol{\phi}_2|\mathbf{y}_0, \mathcal{M})P(\mathbf{y}_0|\mathcal{M})}{P(\mathbf{y}_0|\boldsymbol{\phi}_2, \mathcal{M})}\ , \end{gathered} \tag{11}$$

where the first equality derives from the independence of the speaker factors, and the second equality from Bayes rule.
Substituting (10) and (11) in (8) we get:

$$\lambda = \log \frac{P(\mathbf{y}_0|\boldsymbol{\phi}_1, \mathcal{M})\ P(\mathbf{y}_0|\boldsymbol{\phi}_2, \mathcal{M})}{P(\mathbf{y}_0|\mathcal{M})\ P(\mathbf{y}_0|\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \mathcal{M})}\ . \tag{12}$$

Using (4) and (6), and selecting $\mathbf{y}_0 = \mathbf{0}$, we finally get the log–likelihood ratio:

$$\begin{aligned} \lambda = \frac{1}{2}(&\log|\tilde{\boldsymbol{\Gamma}}| - \boldsymbol{\gamma}_1{}^T\tilde{\boldsymbol{\Gamma}}\boldsymbol{\gamma}_1 + \log|\tilde{\boldsymbol{\Gamma}}| - \boldsymbol{\gamma}_2{}^T\tilde{\boldsymbol{\Gamma}}\boldsymbol{\gamma}_2 \\ &- \log|\mathbf{B}| + \boldsymbol{\mu}^T\mathbf{B}\boldsymbol{\mu} - \log|\tilde{\boldsymbol{\Lambda}}| + \boldsymbol{\gamma}_{1,2}^T\tilde{\boldsymbol{\Lambda}}\boldsymbol{\gamma}_{1,2})\ , \end{aligned} \tag{13}$$

where, according to (7):

$$\begin{aligned} \tilde{\boldsymbol{\Lambda}} &= (\mathbf{B} + 2\mathbf{W})^{-1} & \tilde{\boldsymbol{\Gamma}} &= (\mathbf{B} + \mathbf{W})^{-1} \\ \boldsymbol{\gamma}_{1,2} &= \mathbf{B}\boldsymbol{\mu} + \mathbf{W}(\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2) & \boldsymbol{\gamma}_i &= \mathbf{B}\boldsymbol{\mu} + \mathbf{W}\boldsymbol{\phi}_i\ . \end{aligned} \tag{14}$$

Collecting in a constant $\tilde{k}$ all the terms in the sum that are not a function of $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_{1,2}$, (13) can be rewritten as:

$$\lambda\ = \frac{1}{2}\left(\tilde{k} + \boldsymbol{\gamma}_{1,2}^T\tilde{\boldsymbol{\Lambda}}\boldsymbol{\gamma}_{1,2} - \boldsymbol{\gamma}_1{}^T\tilde{\boldsymbol{\Gamma}}\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2{}^T\tilde{\boldsymbol{\Gamma}}\boldsymbol{\gamma}_2\ \right) \tag{15}$$

with

$$\tilde{k} = 2\log|\tilde{\boldsymbol{\Gamma}}| - \log|\tilde{\boldsymbol{B}}| - \log|\tilde{\boldsymbol{\Lambda}}| + \boldsymbol{\mu}^T\mathbf{B}\boldsymbol{\mu} \tag{16}$$

Substituting (14) in (15) to make the role of the two i–vectors in the log–likelihood ratio computation explicit, we obtain the score:

$$\begin{aligned} s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = & \\ & \frac{1}{2}\Big((\mathbf{B}\boldsymbol{\mu} + \mathbf{W}(\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2))^T \tilde{\boldsymbol{\Lambda}} (\mathbf{B}\boldsymbol{\mu} + \mathbf{W}(\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2)) \\ & -(\mathbf{B}\boldsymbol{\mu} + \mathbf{W}\boldsymbol{\phi}_1)^T \tilde{\boldsymbol{\Gamma}} (\mathbf{B}\boldsymbol{\mu} + \mathbf{W}\boldsymbol{\phi}_1) \\ & -(\mathbf{B}\boldsymbol{\mu} + \mathbf{W}\boldsymbol{\phi}_2)^T \tilde{\boldsymbol{\Gamma}} (\mathbf{B}\boldsymbol{\mu} + \mathbf{W}\boldsymbol{\phi}_2) + \tilde{k} \Big) \ , \end{aligned} \tag{17}$$

which can be rewritten as:

$$\begin{aligned} s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \ = \ & \boldsymbol{\phi}_1^T \boldsymbol{\Lambda} \boldsymbol{\phi}_2 + \boldsymbol{\phi}_2^T \boldsymbol{\Lambda} \boldsymbol{\phi}_1 + \boldsymbol{\phi}_1^T \boldsymbol{\Gamma} \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2^T \boldsymbol{\Gamma} \boldsymbol{\phi}_2 \\ & + (\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2)^T \mathbf{c} + k \ . \end{aligned} \tag{18}$$

Thus, the speaker verification score is a quadratic function of the i–vector pair in a trial, where the original model parameters are related to $\boldsymbol{\Lambda}, \boldsymbol{\Gamma}, \mathbf{c}$ and k according to:

$$\begin{aligned} & \boldsymbol{\Lambda} = \tfrac{1}{2}\mathbf{W}^T \tilde{\boldsymbol{\Lambda}} \mathbf{W} && \boldsymbol{\Gamma} = \tfrac{1}{2}\mathbf{W}^T (\tilde{\boldsymbol{\Lambda}} - \tilde{\boldsymbol{\Gamma}}) \mathbf{W} \\ & \boldsymbol{c} = \mathbf{W}^T (\tilde{\boldsymbol{\Lambda}} - \tilde{\boldsymbol{\Gamma}}) \mathbf{B}\boldsymbol{\mu} && k = \tilde{k} + \tfrac{1}{2}\Big((\mathbf{B}\boldsymbol{\mu})^T (\tilde{\boldsymbol{\Lambda}} - 2\tilde{\boldsymbol{\Gamma}}) \mathbf{B}\boldsymbol{\mu} \Big) \ . \end{aligned} \tag{19}$$

Since the two–covariance model is a particular case of the PLDA approach, where the dimensionality of the speaker and channel spaces is full, its parameters, $\mathbf{B}$,$\mathbf{W}$, and $\boldsymbol{\mu}$ can be trained by means of the same EM algorithm that has been used for PLDA [4].

Another derivation, based on the two–covariance model leading to the same formulation has been illustrated in [15].

D. Expanded vector linear classifier

To demonstrate that the log–likelihood ratio score $s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ of (18) can be computed as a dot–product in an i–vector pairs expanded space, we recall that the computation of the bilinear form $\mathbf{x}^T \mathbf{A} \mathbf{y}$ can be expressed in terms of the Frobenius inner product as $\mathbf{x}^T \mathbf{A} \mathbf{y} = \langle \mathbf{A}, \mathbf{x}\mathbf{y}^T \rangle = \mathrm{vec}(\mathbf{A})^T \mathrm{vec}(\mathbf{x}\mathbf{y}^T)$, where $\mathrm{vec}(\cdot)$ is the operator that stacks the columns of a matrix into a vector and $\langle \mathbf{A}, \mathbf{B} \rangle$ denotes the dot–product between matrices $\mathbf{A}$ and $\mathbf{B}$ Hence, the expression for the speaker verification log-likelihood ratio score (18) can be rewritten as:

$$\begin{aligned} s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = & \langle \boldsymbol{\Lambda}, \boldsymbol{\phi}_1 \boldsymbol{\phi}_2^T + \boldsymbol{\phi}_2 \boldsymbol{\phi}_1^T \rangle + \langle \boldsymbol{\Gamma}, \boldsymbol{\phi}_1 \boldsymbol{\phi}_1^T + \boldsymbol{\phi}_2 \boldsymbol{\phi}_2^T \rangle \\ & + \mathbf{c}^T (\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2) + k \ . \end{aligned} \tag{20}$$

By stacking the parameters as:

$$\mathbf{w} = \begin{bmatrix} \mathrm{vec}(\boldsymbol{\Lambda}) \\ \mathrm{vec}(\boldsymbol{\Gamma}) \\ \mathbf{c} \\ k \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{\boldsymbol{\Lambda}} \\ \mathbf{w}_{\boldsymbol{\Gamma}} \\ \mathbf{w}_{\mathbf{c}} \\ \mathbf{w}_k \end{bmatrix} \tag{21}$$

and expanding an i-vector pair as:

$$\varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \begin{bmatrix} \mathrm{vec}(\boldsymbol{\phi}_1 \boldsymbol{\phi}_2^T + \boldsymbol{\phi}_2 \boldsymbol{\phi}_1^T) \\ \mathrm{vec}(\boldsymbol{\phi}_1 \boldsymbol{\phi}_1^T + \boldsymbol{\phi}_2 \boldsymbol{\phi}_2^T) \\ \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \varphi_{\boldsymbol{\Lambda}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ \varphi_{\boldsymbol{\Gamma}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ \varphi_{\mathbf{c}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ \varphi_k(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \end{bmatrix} \tag{22}$$

$s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ can be written as the dot–product of a vector of weights $\mathbf{w}$ (the model hyper–parameters) and an expanded vector $\varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ representing a trial:

$$\begin{aligned} s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \; & S_{\boldsymbol{\Lambda}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) + S_{\boldsymbol{\Gamma}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ & + S_{\mathbf{c}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) + S_k(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ = \; & \mathbf{w}_{\boldsymbol{\Lambda}}^T \varphi_{\boldsymbol{\Lambda}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) + \mathbf{w}_{\boldsymbol{\Gamma}}^T \varphi_{\boldsymbol{\Gamma}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \; + \\ & \mathbf{w}_{\mathbf{c}}^T \varphi_{\mathbf{c}}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) + \mathbf{w}_k^T \varphi_k(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ = \; & \mathbf{w}^T \varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \; . \end{aligned} \tag{23}$$

E. Taylor approximation of the speaker verification score

In this section, we show that it is possible to discriminate between same–speaker and different–speaker trials, without having to explicitly model the distributions of i-vectors, i.e., without making reference to the two-covariance model.

The same expansion $\varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ defined in (22) can be obtained as a second order Taylor expansion of a speaker verification score. Let's assume that the speaker verification score is an analytic function $s(\boldsymbol{\Phi})$ of the i–vector pair $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$, invariant to i–vector swapping, i.e., $s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = s(\boldsymbol{\phi}_2, \boldsymbol{\phi}_1)$. The Taylor expansion for s, around a point $\hat{\boldsymbol{\Phi}}$, is:

$$s(\boldsymbol{\Phi}) = \sum_{k=0}^{+\infty} \frac{\left(\left(\boldsymbol{\Phi} - \hat{\boldsymbol{\Phi}} \right) \cdot \boldsymbol{\nabla} \right)^k s|_{\hat{\boldsymbol{\Phi}}}}{k!} \; , \tag{24}$$

where $\boldsymbol{\nabla}$ is the vector of differential operators

$$\boldsymbol{\nabla} = \left(\frac{\partial}{\partial \Phi_1}, \ldots, \frac{\partial}{\partial \Phi_d} \right) \; , \tag{25}$$

and d is the dimension of the i–vector pair.

In order to preserve the symmetry of the Taylor polynomials without having to further constrain the score function we consider Taylor series around symmetric points, i.e., $\hat{\boldsymbol{\Phi}} = (\boldsymbol{\phi}_0, \boldsymbol{\phi}_0)$ for some $\boldsymbol{\phi}_0$. In particular, let's consider the second order Taylor expansion for $s(\boldsymbol{\Phi})$ around the point $\hat{\boldsymbol{\Phi}} = \mathbf{0}$:

$$s(\boldsymbol{\Phi}) = s(\hat{\boldsymbol{\Phi}}) + (\boldsymbol{\Phi} \cdot \boldsymbol{\nabla} s|_{\hat{\boldsymbol{\Phi}}}) + \boldsymbol{\Phi}^T (\mathbf{H}(s)|_{\hat{\boldsymbol{\Phi}}}) \boldsymbol{\Phi} \; , \tag{26}$$

where $\mathbf{H}(s)$ is the Hessian of function $s(\boldsymbol{\Phi})$. If we define:

$$\begin{aligned} \mathbf{H}(s)|_{\hat{\boldsymbol{\Phi}}} &= \begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Lambda} \\ \boldsymbol{\Lambda} & \boldsymbol{\Gamma} \end{bmatrix} \\ \boldsymbol{\nabla} s|_{\hat{\boldsymbol{\Phi}}} &= [\, \mathbf{c} \quad \mathbf{c} \,] \\ s(\hat{\boldsymbol{\Phi}}) &= k \; , \end{aligned} \tag{27}$$

with a symmetric $\boldsymbol{\Lambda}$, we obtain the same score formulation as in (18). It is worth noting that the structure imposed by (27) arises naturally from the symmetry of the score function $s(\boldsymbol{\Phi})$ and from the symmetry of the expansion point $\hat{\boldsymbol{\Phi}}$. It does not depend on the particular choice of $\hat{\boldsymbol{\Phi}} = \mathbf{0}$. It is possible to prove (see Appendix A) that, for any choice of a symmetric $\hat{\boldsymbol{\Phi}}$ all Taylor expansion polynomials for $s(\boldsymbol{\Phi})$ at $\hat{\boldsymbol{\Phi}}$ are symmetric, and that the coefficients of the Taylor expansion of $s(\boldsymbol{\Phi})$ at $\hat{\boldsymbol{\Phi}}$ have exactly the structure of (27).

Since the second order Taylor approximation of the scoring function around a symmetric point has the structure described in (18), the pairwise discriminative training approach, which is illustrated in the next section, can be interpreted as a procedure that estimates the parameters of the second order approximation of a good score function, according to the SVM optimization criterion.

IV. Discriminative classifiers

Using the expanded vector $\varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ representing a trial, pairwise discriminative training can be performed by estimating the weights $\mathbf{w}$ in (23). We estimate these weights by means of a linear discriminative classifier,

e.g. a Support Vector Machine. A Support Vector Machine [16], [17], [18] is a binary classifier which estimates the hyperplane that best discriminates two given classes of patterns according to a maximum separation margin criterion. The separation hyperplane is obtained by solving the problem:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\lambda\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1-\zeta_i\mathbf{w}^T\mathbf{x}_i\right) \ , \tag{28}$$

where n is the number of training patterns, $\mathbf{x}_i \in \mathcal{X}$ denotes a (d–dimensional) training pattern with associated label $\zeta_i \in \{-1, +1\}$, and λ is a regularization factor. The second term in this expression is the empirical risk evaluated on the training set, whereas the first term — the squared $L2$ norm of the separating hyperplane $\mathbf{w}$ — is a regularization contribution, which is related to the generalization capability of the model [17]. The regularization factor λ allows tuning the trade–off between the margin and the empirical risk. The latter is the sum of so–called hinge (L1) loss function:

$$l_{L1}(i) = \max(0, 1-\zeta_i\mathbf{w}^T\mathbf{x}_i) \ . \tag{29}$$

The minimization of (28) gives the maximum soft–margin classifier. The SVM is a linear classifier, however, a non–linear classifier can be obtained by means of the so called "kernel trick" [19] where every dot product is replaced by a nonlinear kernel function, or as in our case, by means of a non–linear feature expansion. In fact, the feature mapping (22) defines a linear kernel that is equivalent to a second degree inhomogeneous polynomial kernel:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \left(\mathbf{x}_1^T\mathbf{x}_2 + 1\right)^2 \ , \tag{30}$$

where $\mathbf{x}_1 = [\boldsymbol{\phi}_a\, \boldsymbol{\phi}_b]$ and $\mathbf{x}_2 = [\boldsymbol{\phi}_w\, \boldsymbol{\phi}_z]$ define two different speaker recognition trials. The kernel

$$\begin{aligned} K(\mathbf{x}_1, \mathbf{x}_2) &= K([\boldsymbol{\phi}_a\, \boldsymbol{\phi}_b], [\boldsymbol{\phi}_w\, \boldsymbol{\phi}_z]) \\ &= \left(\boldsymbol{\phi}_a^T\boldsymbol{\phi}_w + \boldsymbol{\phi}_b^T\boldsymbol{\phi}_z + 1\right)^2 \end{aligned} \tag{31}$$

can be rewritten as:

$$\begin{aligned} K(\mathbf{x}_1, \mathbf{x}_2) =& \boldsymbol{\phi}_a^T\boldsymbol{\phi}_w\boldsymbol{\phi}_w^T\boldsymbol{\phi}_a + \boldsymbol{\phi}_b^T\boldsymbol{\phi}_z\boldsymbol{\phi}_z^T\boldsymbol{\phi}_b + \\ & 2\boldsymbol{\phi}_a^T\boldsymbol{\phi}_w\boldsymbol{\phi}_z^T\boldsymbol{\phi}_b + 2\boldsymbol{\phi}_a^T\boldsymbol{\phi}_w + 2\boldsymbol{\phi}_b^T\boldsymbol{\phi}_z + 1 \\ =& \langle\boldsymbol{\phi}_a\boldsymbol{\phi}_a^T, \boldsymbol{\phi}_w\boldsymbol{\phi}_w^T\rangle + \langle\boldsymbol{\phi}_b\boldsymbol{\phi}_b^T, \boldsymbol{\phi}_z\boldsymbol{\phi}_z^T\rangle + \\ & 2\langle\boldsymbol{\phi}_a\boldsymbol{\phi}_b^T, \boldsymbol{\phi}_w\boldsymbol{\phi}_z^T\rangle + 2\boldsymbol{\phi}_a^T\boldsymbol{\phi}_w + 2\boldsymbol{\phi}_b^T\boldsymbol{\phi}_z + 1 \ . \end{aligned} \tag{32}$$

Defining the feature mapping:

$$\tilde{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \mathrm{vec}([\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ 1][\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ 1]^T) \sim \begin{bmatrix} \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_2^T) \\ \mathrm{vec}(\boldsymbol{\phi}_2\boldsymbol{\phi}_1^T) \\ \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_1^T) \\ \mathrm{vec}(\boldsymbol{\phi}_2\boldsymbol{\phi}_2^T) \\ \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \boldsymbol{\phi}_2 \\ 1 \end{bmatrix} \ , \tag{33}$$

where $\sim$ is used to denote equivalence of vectors ignoring the order of their elements, we can conclude that the kernel $K(\mathbf{x}_1, \mathbf{x}_2)$ is the dot–product of two expanded vectors:

$$\begin{aligned} K(\mathbf{x}_1, \mathbf{x}_2) =& \langle[\boldsymbol{\phi}_a\boldsymbol{\phi}_b 1][\boldsymbol{\phi}_a\boldsymbol{\phi}_b 1]^T, [\boldsymbol{\phi}_w\boldsymbol{\phi}_z 1][\boldsymbol{\phi}_w\boldsymbol{\phi}_z 1]^T\rangle \\ =& \tilde{\varphi}(\boldsymbol{\phi}_a\boldsymbol{\phi}_b)^T\tilde{\varphi}(\boldsymbol{\phi}_w\boldsymbol{\phi}_z) \ . \end{aligned} \tag{34}$$

Looking at the log–likelihood in (18) and halving its (unknown) parameter $\mathbf{c}$ as $\tilde{\mathbf{c}} = \mathbf{c}/2$, so that the linear term of the log–likelihood becomes $2\tilde{\mathbf{c}}^T(\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2)$, the feature expansion given in (22) becomes:

$$\varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \begin{bmatrix} \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_2^T + \boldsymbol{\phi}_2\boldsymbol{\phi}_1^T) \\ \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_1^T + \boldsymbol{\phi}_2\boldsymbol{\phi}_2^T) \\ 2\ (\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2) \\ 1 \end{bmatrix} \tag{35}$$

and it is easy to verify that the two expansions:

$$\varphi(\boldsymbol{\phi}_a, \boldsymbol{\phi}_b)^T \varphi(\boldsymbol{\phi}_w, \boldsymbol{\phi}_z) = \tilde{\varphi}(\boldsymbol{\phi}_a, \boldsymbol{\phi}_b)^T \tilde{\varphi}(\boldsymbol{\phi}_w, \boldsymbol{\phi}_z) \tag{36}$$

are equivalent, i.e., correspond to the same kernel.

Often, SVM classifiers are trained using a solver of the dual problem, where a Gram matrix needs to be evaluated. The Gram matrix contains the dot–products between every pair of training examples. Since our training examples are i–vector pairs, the size of the Gram matrix — $O(n^4)$ due to the square of n^2 i–vector pairs — would be unacceptably large. Therefore, we train an SVM by solving the primal problem using a general solver (see Section V), and an efficient evaluation of loss function gradient that allow both memory and computational resources to be constrained.

Although the G–PLDA and the pairwise SVM expressions are formally equivalent, an important difference has to be highlighted considering the hyper–parameters that are trained. The parameters estimated in G–PLDA (and two–covariance) model are constrained, due to the positive definiteness constraints of their covariance matrices. In the pairwise discriminative training approach, instead, no parameter constraints are imposed, except for the ones arising from the regularization of the optimization function. Thus, the latter approach is more flexible and does not make a priori assumptions about the i–vector distribution.

It is also worth noting that the same task can be performed by Logistic Regression (LR), another widely used linear classifier, which allows estimating class posterior probabilities given a set of patterns [18]. Normalizing the loss function of LR by the number of patterns n, and including a regularization factor $\frac{\lambda}{2}\|\mathbf{w}\|^2$, the regularized LR objective function $f_{LR}(\hat{\mathbf{w}})$ is:

$$f_{LR}(\hat{\mathbf{w}}) = \frac{\lambda}{2}\|\hat{\mathbf{w}}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-\zeta_i \hat{\mathbf{w}}^T \hat{\mathbf{x}}_i}\right) \ , \tag{37}$$

which is similar to the SVM objective function. SVM and LR optimization can be seen as the solution of a particular instance of the unconstrained convex regularized risk minimization problem:

$$E(\mathbf{w}) = \arg\min_{\mathbf{w}} \frac{1}{2}\lambda \left\|\mathbf{w}\right\|^2 + \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x_i}, \zeta_i) \tag{38}$$

with loss function

$$\ell_{L1}(i) = \max(0, 1 - \zeta_i \mathbf{w}^T \mathbf{x}_i) \tag{39}$$

and

$$\ell_{LR}(i) = \log\left(1 + e^{-\zeta_i \hat{\mathbf{w}}^T \hat{\mathbf{x}}_i}\right) \ , \tag{40}$$

respectively. The SVM optimizes the margin separation between the classes, whereas LR minimizes the cross–entropy error function.

In the following we will illustrate our solutions and report results for the SVM classifier, but the same considerations apply to LR, just changing the loss function. The results of some experiments comparing these two discriminative classifiers have been reported in [15].

V. Pairwise SVM training

Since our training patterns are all possible pairs of i–vectors in the training set, their number grows as $O(n^2)$. The feature mapping described in Section III-D produces mapped features having $O(d^2)$ components, thus the global dataset size would be $O(n^2 d^2)$. Caching the complete kernel matrix is impractical even for relatively small sized datasets because it would require $O(n^4)$ memory. In [20] we have shown that SVM training of the i–vector pairs

by means of a dual solver requires either keeping in memory the complete dataset of mapped features ($O(n^2d^2)$), or mapping the feature on–line, with a complexity $O(n^2d^2)$ for each iteration. Since in our experiments $d = 400$ and n is approximately 20000, a standard dual solver approach is not viable.
Training is feasible, instead, by using a primal solver because, as we show in Section VI, it is possible to efficiently evaluate the loss function and its gradient with respect to $\mathbf{w}$ over the set of all training trials in $O(n^2d+nd^2)$ time, without the need to expand the i–vectors. Due to the small size of the i–vectors, the dataset of training utterance can easily be loaded in main memory. The evaluation of loss functions and gradients in these algorithms requires matrix–by–matrix multiplications of large matrices ($n\times n$), however it is not necessary to store the complete matrices in main memory because the computations can be performed through block decomposition of the matrices.

An analysis of large-scale SVM training algorithms suited to speaker recognition tasks [20] allowed us to select, among the primal solvers, the Bundle Methods for Regularized Risk Minimization (BMRM) [21], [22], which offer a general and easily extensible framework for solving convex unconstrained regularized risk minimization problems. In particular, we trained our SVM using the Optimized Cutting Plane Algorithm (OCAS) approach proposed in [23], [22], which is an extension to BMRM that shows better and smoother convergence properties.
An important advantage of these methods is that they do not require the loss function to be differentiable in the whole domain.

VI. Efficient score and gradient computation

Using the OCAS technique, the SVM parameters $\mathbf{w}$ are optimized by evaluating the loss function and a sub–gradient of its error function (38):

$$\nabla E(\mathbf{w}) = \frac{1}{n}\sum_{\boldsymbol{\phi}_i,\boldsymbol{\phi}_j} \frac{\partial \ell(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)}{\partial s(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)} \frac{\partial s(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)}{\partial \mathbf{w}} + \lambda \mathbf{w} \ . \tag{41}$$

The use of sub–gradients for optimization [24] is necessary because the hinge loss function is not differentiable everywhere. A sub–gradient for the SVM hinge loss function is:

$$\frac{\partial \ell_{L1}(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)}{\partial s(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)} = \begin{cases} 0 & \text{if } \zeta_{i,j}\ s(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j) \geq 1 \\ -\zeta_{i,j} & \text{otherwise ,} \end{cases} \tag{42}$$

where $\zeta_{i,j} \in \{-1,+1\}$ is the label of the i–vector pair $(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)$. The derivative of the score with respect to the classifier parameters is simply the expanded trial vector:

$$\frac{\partial s(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \mathbf{w}^T \varphi(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j) = \varphi(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j) \ . \tag{43}$$

The evaluation of the loss function and its gradient requires, in principle, a sum over all the expanded i–vector pairs in the training set. Since their number is n^2, which can easily reach the order of hundred of millions for typical training sets, these evaluations would be not effective or even feasible because the complexity would be $O(n^2d^2)$. In the next section, however, we show that these computations can be done without an explicit full expansion of all the i-vector pairs, with a complexity that reduces to $O(n^2d + nd^2)$.

A. Fast scoring

In order to obtain effectively the loss contributions of all training pairs, we need a fast procedure for computing the scores of all the training i–vector pairs, obtaining the matrix of the scores of every i–vector against each other.

Given a trained classifier, a verification score for a trial pair can be computed by means of the expanded vector $\varphi(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)$ and the dot–product in (22) and (23). However, a much more efficient solution in terms of memory and computation can be obtained using (18). In particular, let $\mathbf{D} = [\boldsymbol{\phi}_1\boldsymbol{\phi}_2 \ldots \boldsymbol{\phi}_n]$ be a matrix including n stacked i-vectors, and let $\mathbf{S}_{\Theta i,j} = \mathbf{S}_\Theta(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)$ denote the score matrix for all possible trials related to component Θ of $\mathbf{w}$,

where $\Theta \in \{\mathbf{\Lambda}, \mathbf{\Gamma}, \mathbf{c}, k\}$. From (23) and (18) the score matrices can be evaluated as:

$$\begin{aligned} S_{\mathbf{\Lambda}}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) &= \boldsymbol{\phi}_i^T \mathbf{\Lambda} \boldsymbol{\phi}_j + \boldsymbol{\phi}_j^T \mathbf{\Lambda} \boldsymbol{\phi}_i \Rightarrow \mathbf{S}_{\mathbf{\Lambda}} = 2\ \mathbf{D}^T \mathbf{\Lambda}\, \mathbf{D} \\ S_{\mathbf{\Gamma}}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) &= \boldsymbol{\phi}_i^T \mathbf{\Gamma} \boldsymbol{\phi}_i + \boldsymbol{\phi}_j^T \mathbf{\Gamma} \boldsymbol{\phi}_j \Rightarrow \mathbf{S}_{\mathbf{\Gamma}} = \tilde{\mathbf{S}_{\mathbf{\Gamma}}} + \tilde{\mathbf{S}_{\mathbf{\Gamma}}}^T \\ S_c(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) &= \mathbf{c}^T (\boldsymbol{\phi}_i + \boldsymbol{\phi}_j) \qquad \Rightarrow \mathbf{S}_c = \tilde{\mathbf{S}_c} + \tilde{\mathbf{S}_c}^T \\ S_k(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) &= k \qquad\qquad\qquad\ \Rightarrow \mathbf{S}_k = k \cdot \mathbf{1}\ , \end{aligned} \tag{44}$$

where

$$\tilde{\mathbf{S}_{\mathbf{\Gamma}}} = [\underbrace{d_{\mathbf{\Gamma}} \dots d_{\mathbf{\Gamma}}}_{n}] \qquad \tilde{\mathbf{S}_c} = [\underbrace{d_c \dots d_c}_{n}] \tag{45}$$

and

$$d_{\mathbf{\Gamma}} = \mathrm{diag}(\mathbf{D}^T \mathbf{\Gamma} \mathbf{D}) \qquad d_c = \mathbf{D}^T \mathbf{c}\ . \tag{46}$$

The operator diag returns the diagonal of a matrix as a column vector, and $\mathbf{1}$ is an $n \times n$ matrix of ones. No explicit expansion of i-vectors is therefore necessary for this evaluation.

B. Loss function evaluation

Denoting by $\mathbf{S} = \mathbf{S}_{\mathbf{\Lambda}} + \mathbf{S}_{\mathbf{\Gamma}} + \mathbf{S}_{\mathbf{c}} + \mathbf{S}_k$ the sum of the partial score matrices, the SVM loss function can be obtained as:

$$\begin{aligned} \boldsymbol{\ell}_{L1}(\mathbf{D}, \mathbf{Z}) &= \sum_{i,j} \max(0, 1 - \zeta_{i,j} \mathbf{w}^T \varphi(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) \\ &= \langle \mathbf{1}, \max(\mathbf{0}, \mathbf{1} - (\boldsymbol{Z} \circ \mathbf{S}) \rangle\ , \end{aligned} \tag{47}$$

where $\mathbf{0}$ is an $n \times n$ matrix of zeros, $\boldsymbol{Z}$ is the $n \times n$ matrix of the trial labels $\zeta_{i,j}$ for each i–vector pair $(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$, and $\circ$ is the element-wise matrix multiplication operator.

C. Gradient evaluation

The sub–gradient of the loss function can be evaluated from its derivative with respect to the m-th dimension of $\mathbf{w}$ as:

$$\begin{aligned} \frac{\partial \boldsymbol{\ell}}{\partial w_m} &= \sum_{i,j} \frac{\partial \ell(w, (\boldsymbol{\phi}_i, \boldsymbol{\phi}_j), \zeta_{i,j})}{\partial (w^T \varphi(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j))} \frac{\partial w^T \varphi(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)}{\partial w_m} \\ &= \sum_{i,j} g_{i,j} \frac{\partial s_{i,j}}{\partial w_m} = \sum_{i,j} g_{i,j} \varphi(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)_m\ , \end{aligned} \tag{48}$$

where $g_{i,j}$ is the derivative of the hinge loss function with respect to the score $s_{i,j} = \mathbf{w}^T \varphi(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$:

$$g_{i,j} = \begin{cases} 0 & \text{if } \zeta_{i,j} s_{i,j} \geq 1 \\ -\zeta_{i,j} & \text{otherwise}\ . \end{cases} \tag{49}$$

Considering the i–vector expansion (22), the loss function gradient (48) can be written as:

$$\nabla \boldsymbol{\ell} = \begin{bmatrix} \nabla_{\mathbf{\Lambda}}\ \boldsymbol{\ell} \\ \nabla_{\mathbf{\Gamma}}\ \boldsymbol{\ell} \\ \nabla_{\mathbf{c}}\ \boldsymbol{\ell} \\ \nabla_k\ \boldsymbol{\ell} \end{bmatrix} = \begin{bmatrix} \mathrm{vec}\left(\sum_{i,j} g_{i,j} \left(\boldsymbol{\phi}_i \boldsymbol{\phi}_j^T + \boldsymbol{\phi}_j \boldsymbol{\phi}_i^T \right) \right) \\ \mathrm{vec}\left(\sum_{i,j} g_{i,j} \left(\boldsymbol{\phi}_i \boldsymbol{\phi}_i^T + \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \right) \right) \\ \sum_{i,j} g_{i,j} \left(\boldsymbol{\phi}_i + \boldsymbol{\phi}_j \right) \\ \sum_{i,j} g_{i,j} \end{bmatrix}\ . \tag{50}$$

Defining $\mathbf{G}$ the matrix of the elements $g_{i,j}$, and taking into account that it is symmetric, the terms of the sub–gradient of the loss function, related to a component Θ of $\mathbf{w}$, can be expressed in terms of dot–products and

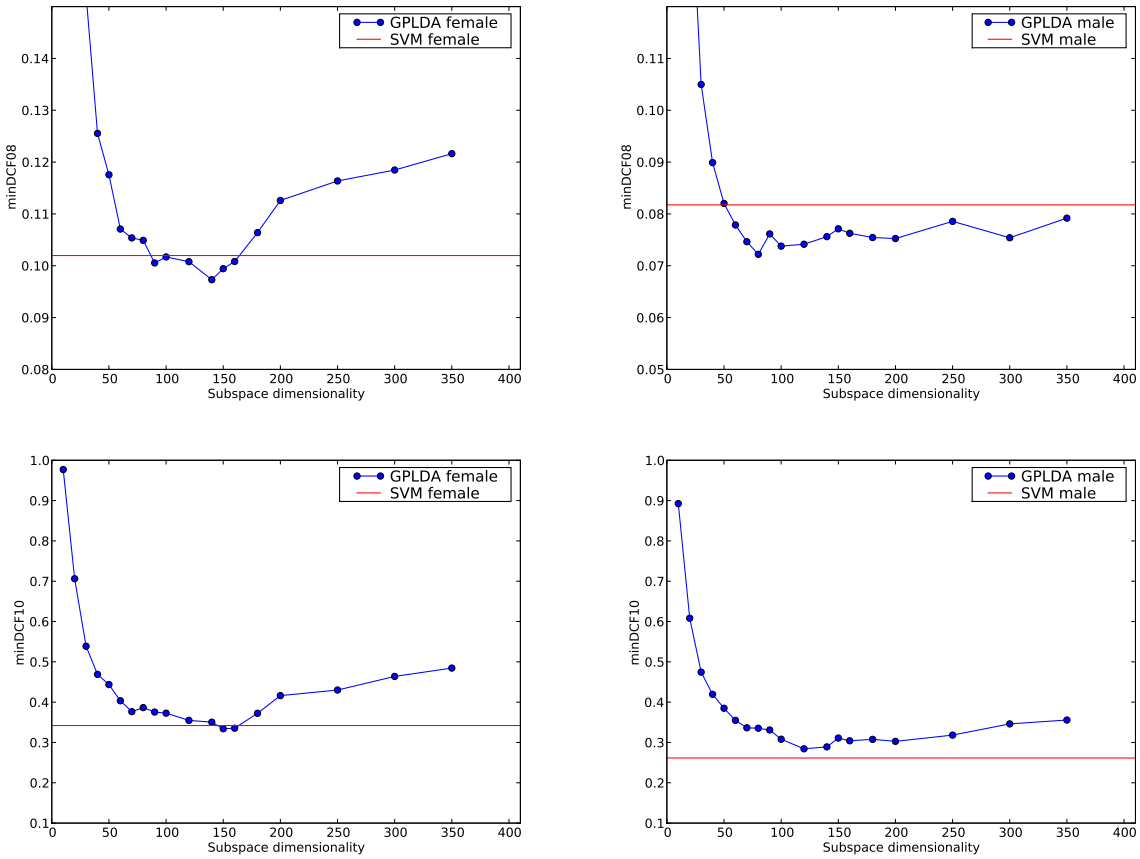


Fig. 1: minDFC08 and minDCF10 as a function of the speaker subspace dimensionality for the female and male speakers

element–wise matrix products as:

$$\nabla \ell = \begin{bmatrix} 2 \text{ vec}\left(\mathbf{D}\mathbf{G}\mathbf{D}^T\right) \\ 2 \text{ vec}\left(\left[\mathbf{D} \circ (\mathbf{1_A}\ \mathbf{G})\right]\mathbf{D}^T\right) \\ 2\ \left[\mathbf{D} \circ (\mathbf{1_A}\ \mathbf{G})\right]\ \mathbf{1_B} \\ \mathbf{1}_B^T\ \mathbf{G}\ \mathbf{1_B} \end{bmatrix}, \tag{51}$$

where $\mathbf{1_A}$ is a $M \times n$ matrix of ones (M is the i–vector dimension) and $\mathbf{1_B}$ is a size n column vector of ones. Again, no explicit expansion of i-vectors is necessary for this evaluation.

TABLE I: Comparison of the performance of G–PLDA with and without i–vector length normalization and PSVM

i–vector length normalization	System	Female			Male		
		EER (%)	minDCF08	minDCF10	EER (%)	minDCF08	minDCF10
no	G–PLDA	3.51	0.15	0.39	2.28	0.12	0.43
yes	G–PLDA	2.10	0.10	0.35	1.24	0.07	0.28
no	PSVM	2.21	0.10	0.34	1.96	0.08	0.26

D. Estimation of the regularization factor

Training a risk minimization problem (38) entails the selection of an appropriate value for the regularization factor λ. Different approaches have been proposed to estimate a good factor, such as cross–validation, or fitting

TABLE II: EER and minDCFs for PSVM on SRE2010 tests with 400 and 600 dimension i–vectors

i–vector type	Model	Gender	Female			Male		
		System	EER	minDCF08	minDCF10	EER	minDCF08	minDCF10
GD	GD	400 GD	2.21 %	0.109	0.360	1.73 %	0.081	0.303
GI	GD	400 PGI	2.49 %	0.115	0.369	1.84 %	0.084	0.298
GI	GI	400 GI	2.51 %	0.115	0.382	1.82 %	0.087	0.309
GD	GD	600 GD	2.32 %	0.106	0.342	1.76 %	0.077	0.290
GI	GD	600 PGI	2.59 %	0.103	0.358	1.82 %	0.082	0.274
GI	GI	600 GI	2.51 %	0.108	0.383	1.80 %	0.078	0.307

the models for all possible regularization factors [25]. After a few cross-validation search strategies were tried, we found that the simple heuristic factor proposed as the default regularization parameter in SVM$^{\text{Light}}$ [26] is sufficient to produce accurate models. It has the advantage that it can be easily computed from the training data as:

$$C = \frac{1}{n\lambda} = \left(\frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i\| \right)^{-2} , \tag{52}$$

where $\mathbf{x}_i$ is one of the n patterns in the training set $\{\mathcal{X}\}$. In our approach a pattern $\mathbf{x}_i$ is an i–vector pair. Looking at (34) and (31), by replacing $\boldsymbol{\phi}_w$ and $\boldsymbol{\phi}_z$ with $\boldsymbol{\phi}_a$ and $\boldsymbol{\phi}_b$, respectively, the norm of the expanded features $\varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ for the i–vector pair $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ can be computed as:

$$\|\varphi(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)\| = \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2^T \boldsymbol{\phi}_2 + 1 \ . \tag{53}$$

Thus the regularization parameter λ can be set so that:

$$\begin{aligned} \frac{1}{n\lambda} &= \left(\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\varphi(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)\| \right)^{-2} \\ &= \left(\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\boldsymbol{\phi}_i^T \boldsymbol{\phi}_i + \boldsymbol{\phi}_j^T \boldsymbol{\phi}_j + 1 \right) \right)^{-2} \\ &= \left(1 + \frac{2}{n} \sum_{i=1}^{n} \|\boldsymbol{\phi}_i\|^2 \right)^{-2} . \end{aligned} \tag{54}$$

VII. Experimental Results

The i-vector extractor used for the first set of experiments is based on 60-dimensional cepstral features and a 2048-component full covariance GMM. The UBM and i-vector extractor are trained on NIST SRE 2004, 2005 and 2006, Switchboard and Fisher data. The PLDA systems and discriminative classifiers have been trained using i-vectors with dimension d =400 or $d = 600$, respectively, extracted from NIST SRE 2004, NIST SRE 2005, NIST SRE 2006, Switchboard II Phases 2 and 3, and Switchboard Cellular Parts 1 and 2.

Table I presents the results for the extended condition 5 (tel–tel) from NIST SRE 2010 evaluation in terms of percent Equal Error Rate and normalized minimum Detection Cost Function (minDCF) as defined by NIST for SRE08 and SRE10 evaluations [11].

The system denoted as G–PLDA without length normalization is based on a generatively trained PLDA model with a 120–dimensional speaker variability subspace, and full channel variability subspace. For the system denoted as G–PLDA with length normalization, which is our reference, we perform in sequence within-class covariance normalization [27] and length normalization of the i–vectors [13]. This configuration was found to give the best minDCF10, which was the primary performance measure in NIST SRE 2010 evaluation focusing on low false alarm rates.

In the Pairwise SVM (PSVM) system, the lack of normalization of the i-vector dimensions would affect the regularization term $\frac{1}{2}\lambda\|\mathbf{w}\|^2$ in the SVM objective function (28). Thus, to make SVM regularization effective, we normalize the i–vectors so that they have identity within-speaker covariance matrix.

In these conditions, the behavior of the two systems is similar (and much better than G–PLDA without i–vector length normalization [13]).

Since the G–PLDA with length-normalized i–vectors performs slightly better than PSVM, we performed another set of experiments to assess the effects of size of the speaker variability subspace on G–PLDA accuracy. Figure 1 shows the minDFC08 and minDCF10 as a function of the speaker variability subspace dimension for the female and male speakers separately. We can observe in these figures that the dimension of the speaker variability subspace must be carefully tuned because it affects system performance. No tuning is necessary in our pairwise SVM models because we always estimate full–rank $\mathbf{\Lambda}$ and $\mathbf{\Gamma}$ matrices.

A. Gender–independent pairwise SVM

State–of–the–art text–independent speaker recognition systems are designed to achieve best performance when the gender label is known both at training and testing time. Gender information, however, is not available in a number of real applications. Although the speaker gender can be estimated from the trial data, this preliminary classification is a potential source of accuracy degradation.

The interpretation of pairwise discriminative training illustrated in Section III-E provides the rationale for a straightforward approach to gender–independent pairwise discriminative training. If we consider the most elaborated generative models, such as Heavy-Tailed [4] or Mixtures of PLDA [28], we can notice that they differ only in the formal expression of their log–likelihood ratio score function. Since in pairwise SVM training we directly optimize a second order approximation of a good score function, a gender–independent SVM can be implemented by training a single system with pooled gender i–vectors, without the need for gender labels both in training and in testing. The gender prior is implicitly built into the SVM solution via the proportions of males and females in the training data, thus some care might be required in case of very unbalanced male and female training sets. The PLDA mixture solution has the advantage (at least in principle) that the user can specify this prior externally at run-time, if the user knows, for example, that females may be scarce in a certain application. In practice however, calibration of the gender likelihoods relative to the prior may cause the user's prior not to have much effect. A gender–independent system has two benefits: a larger amount of training data can be used for off-line estimation of the UBM and of the speaker and inter–session sub–spaces, moreover its models require less memory and computation during testing. Memory is saved because there is no need to keep separate gender models, and unless the knowledge of the gender is a–priori known, a gender detector is needed for a gender–dependent system.

It is worth noting that from the experiments with GD systems, reported in Table I, we know that the pairwise SVM system and the PLDA systems using the same GD i–vectors give comparable performance. We did not train, however, a GI PLDA system using GI i–vectors because the results given [28] for similar telephone tests, show that it is necessary to use mixtures of PLDA models to reach the performance of a GD PLDA system trained with the same GI i–vectors. We focused, thus, only on pairwise SVM systems using GI i–vectors, to assess their performance in a fully GI speaker verification task. In particular, we trained three types of PSVM systems using i–vectors of 400 and 600 dimensions, respectively:

- a fully gender–dependent (GD) system, where both i–vector extraction and SVM training is gender–dependent,
- a partially gender–independent (PGI) system, where the i–vectors are gender–independent, whereas two SVMs are trained using GD segments,
- a totally gender–independent (GI) system, where both i–vector extraction and SVM training is performed without using gender labels.

For GD and PGI systems gender labels are provided at test time, while for the GI system no gender information is used to score the trials.

The results for these models, reported in Table II on the same extended tel-tel SRE10 evaluation set[1], show that a fully GI system, using both 400 or 600 GI i–vectors, gives comparable performance to a partially gender independent

[1]The GD results of Table II are different with respect to the the ones given in Table I because the list for training matrix T included two additional datasets: Part 1 and 2 of the Fisher English Corpus.

system, which needs the gender labels at test time, and is competitive with the more expensive GD models, which of course not only use GD models but also GD i–vectors. Thus, the relative loss of performance observed with respect to the GD systems is due to the use of GI i–vectors, not to model deficiency.

VIII. Conclusions

In this work we presented a novel framework for discriminative training of speaker verification systems, where a trial is represented, as in the PLDA approach, by an i–vector pair, and the task is discrimination between same–speaker and different–speaker classes. This pairwise SVM approach provides a more natural paradigm to speaker verification compared to the classical one–vs–all discriminative training. We showed that this technique has strong connections with the state–of–the–art generative models, but does not need to explicitly model the i–vector distribution. Rather, it can be interpreted as a procedure that estimates the parameters of a second order approximation of a good score function, or simply as a pairwise second degree polynomial kernel classifier in the i–vector pairs space.

We addressed and solved the time and memory issues raised by a naïve quadratic expansion of the i–vector pairs for an efficient computation of the loss function gradients and of the verification scores.

A fully Gender–Independent discriminative system has been trained which achieves, using GI i–vectors, an accuracy comparable to the one offered by similar Gender–Dependent systems, with the advantage of not requiring two separate models nor gender knowledge.

While some issues are still open, for example extensions of the model to deal with more than a pair of utterances or large–scale training, pairwise discriminative training provides models that allow fast scoring of test utterances achieving state–of–the–art performance.

Appendix A

Proposition 1: For $\hat{\mathbf{\Phi}}_0 = (\boldsymbol{\phi}_0, \boldsymbol{\phi}_0)$, all Taylor polynomials of $s(\mathbf{\Phi})$ at $\hat{\mathbf{\Phi}}_0$ are symmetric with respect to i–vector swapping.

Proof: Since s is symmetric, the functions $f(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ and $g(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = s(\boldsymbol{\phi}_2, \boldsymbol{\phi}_1)$ are equal and, therefore, have the same Taylor polynomials for any given order. Let T_p^f, T_p^g and T_p^s denote the p–th order Taylor polynomials for f, g and s, respectively. We have $T_p^s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = T_p^f(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = T_p^g(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = T_p^s(\boldsymbol{\phi}_2, \boldsymbol{\phi}_1)$, thus the Taylor polynomials for $s(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ at $\hat{\mathbf{\Phi}} = (\boldsymbol{\phi}_0, \boldsymbol{\phi}_0)$ are symmetric for any p. ■

Proposition 2: The coefficients of the first and second order Taylor expansion of $s(\mathbf{\Phi})$ at $\hat{\mathbf{\Phi}} = (\phi_0, \phi_0)$ have the symmetric structure given in (27).

Proof: To derive the structure of the Taylor coefficients given in (27), we first consider the Taylor series of s around $\hat{\mathbf{\Phi}} = \mathbf{0}$:

$$\begin{aligned} s(\mathbf{\Phi}) &= \sum_{k=0}^{+\infty} \frac{(\mathbf{\Phi} \cdot \nabla)^k s|_{\mathbf{0}}}{k!} \\ &= k + \boldsymbol{\phi}_1^T \boldsymbol{c}_1 + \boldsymbol{\phi}_2^T \boldsymbol{c}_2 + \boldsymbol{\phi}_1^T \boldsymbol{A} \boldsymbol{\phi}_1 + \boldsymbol{\phi}_1^T \boldsymbol{B} \boldsymbol{\phi}_2 \\ &\quad + \boldsymbol{\phi}_2^T \boldsymbol{C} \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2^T \boldsymbol{D} \boldsymbol{\phi}_2 + \sum_{k=3}^{+\infty} \frac{(\mathbf{\Phi} \cdot \nabla)^k s|_{\mathbf{0}}}{k!} \ . \end{aligned} \tag{55}$$

We can rewrite the first three terms of the series as:

$$\begin{aligned} H(s)|_{\mathbf{0}} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \\ \nabla s|_{\mathbf{0}} &= [\, \mathbf{c}_1 \quad \mathbf{c}_2 \,] \\ s(\mathbf{0}) &= k \ . \end{aligned} \tag{56}$$

From the symmetry of the Hessian it directly follows that $\boldsymbol{A}$ and $\boldsymbol{D}$ are symmetric and that $\boldsymbol{C} = \boldsymbol{B}^T$.

In order to prove that $\boldsymbol{c}_1 = \boldsymbol{c}_2$, $\boldsymbol{A} = \boldsymbol{D}$, and $\boldsymbol{B}$ is also symmetric, we consider the Taylor expansion of s around $\hat{\mathbf{\Phi}}_{\mathbf{0}}$, computed in $\mathbf{\Phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ and in the symmetric point $\bar{\mathbf{\Phi}} = (\boldsymbol{\phi}_2, \boldsymbol{\phi}_1)$. Since s is symmetric, in these two

points the series has the same value. In particular,

$$s(\bar{\boldsymbol{\Phi}}) = k + \boldsymbol{\phi}_2^T \boldsymbol{c}_1 + \boldsymbol{\phi}_1^T \boldsymbol{c}_2 + \boldsymbol{\phi}_2^T \boldsymbol{A} \boldsymbol{\phi}_2 + \boldsymbol{\phi}_2^T \boldsymbol{B} \boldsymbol{\phi}_1 + \boldsymbol{\phi}_1^T \boldsymbol{C} \boldsymbol{\phi}_2 + \boldsymbol{\phi}_1^T \boldsymbol{D} \boldsymbol{\phi}_1 + \sum_{k=3}^{+\infty} \frac{\left(\bar{\boldsymbol{\Phi}} \cdot \nabla\right)^k s|_{\mathbf{0}}}{k!} \ . \tag{57}$$

Since $\boldsymbol{B} = \boldsymbol{C}^T$, we have that $\boldsymbol{\phi}_a^T \boldsymbol{B} \boldsymbol{\phi}_b + \boldsymbol{\phi}_b^T \boldsymbol{C} \boldsymbol{\phi}_a = 2\boldsymbol{\phi}_a^T \boldsymbol{B} \boldsymbol{\phi}_b$ for any $\boldsymbol{\phi}_a, \boldsymbol{\phi}_b$. Therefore, combining (55) and (57) we get:

$$\begin{aligned} & k + \boldsymbol{\phi}_1^T \boldsymbol{c}_1 + \boldsymbol{\phi}_2^T \boldsymbol{c}_2 + \boldsymbol{\phi}_1^T \boldsymbol{A} \boldsymbol{\phi}_1 + 2\boldsymbol{\phi}_1^T \boldsymbol{B} \boldsymbol{\phi}_2 \\ & + \boldsymbol{\phi}_2^T \boldsymbol{D} \boldsymbol{\phi}_2 + \sum_{k=3}^{+\infty} \frac{\left(\boldsymbol{\Phi} \cdot \nabla\right)^k s|_{\mathbf{0}}}{k!} = \\ & k + \boldsymbol{\phi}_2^T \boldsymbol{c}_1 + \boldsymbol{\phi}_1^T \boldsymbol{c}_2 + \boldsymbol{\phi}_2^T \boldsymbol{A} \boldsymbol{\phi}_2 + 2\boldsymbol{\phi}_1^T \boldsymbol{B}^T \boldsymbol{\phi}_2 \\ & + \boldsymbol{\phi}_1^T \boldsymbol{D} \boldsymbol{\phi}_1 + \sum_{k=3}^{+\infty} \frac{\left(\bar{\boldsymbol{\Phi}} \cdot \nabla\right)^k s|_{\mathbf{0}}}{k!} \end{aligned} \tag{58}$$

The equality (58) holds for any choice of $\boldsymbol{\Phi}$ only if all coefficients of the two polynomials are equal, i.e., if $\boldsymbol{c}_1 = \boldsymbol{c}_2$, $\boldsymbol{A} = \boldsymbol{D}$, and $\boldsymbol{B}$ is symmetric.

Finally, consider a generic symmetric point $\hat{\boldsymbol{\Phi}}_{\mathbf{0}} = (\boldsymbol{\phi}_0, \boldsymbol{\phi}_0)$, and let $h(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = s(\boldsymbol{\phi}_1 + \boldsymbol{\phi}_0, \boldsymbol{\phi}_2 + \boldsymbol{\phi}_0)$. The Taylor expansion of h around $\mathbf{0}$ has, by definition, the same coefficients of the Taylor series for s around $\hat{\boldsymbol{\Phi}}_{\mathbf{0}}$. Moreover, h is symmetric, therefore the Taylor coefficients of its second order Taylor polynomial have the same structure as in (27). ■

REFERENCES

[1] L. Burget *et al.*, "Robust speaker recognition over varying channels," in *Johns Hopkins University CLSP Summer Workshop Report*, 2008. Available at http://www.clsp.jhu.edu/workshops/ws08/documents/jhu_report_main.pdf.

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

[3] 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. 31–44, 2000.

[4] P. Kenny, "Bayesian speaker verification with heavy–tailed priors," in *Keynote presentation, Odyssey 2010, The Speaker and Language Recognition Workshop*, 2010. Available at http://www.crim.ca/perso/patrick.kenny/kenny_Odyssey2010.pdf.

[5] N.Brummer, "A farewell to SVM: Bayes factor speaker detection in supervector space," 2006. Available at https://sites.google.com/site/nikobrummer/.

[6] N. Brümmer and E. de Villiers, "The speaker partitioning problem," in *Proc. Odyssey 2010*, pp. 194–201, 2010.

[7] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for inferences about identity," in *Proceedings of 11th International Conference on Computer Vision*, pp. 1–8, 2007.

[8] 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 *Proceedings of ICASSP 2006*, pp. 97–100, 2006.

[9] S. S. R. Vogt, S. Kajarekar, "Discriminant NAP for SVM speaker recognition," in *Proceedings of Odyssey 2008, The Speaker and Language Recognition Workshop*, 2008.

[10] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," in *Technical report CRIM-06/08-13*, 2005.

[11] The NIST Year 2010 Speaker Recognition Evaluation Plan, Available at http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf.

[12] N. Brümmer, L. Burget, P. Kenny, P. Matějka, E. de Villiers, M. Karafiát, M. Kockmann, O. Glembek, O. Plchot, D. Baum, and M. Senoussauoi, "ABC system description for NIST SRE 2010," in *Proc. NIST 2010 Speaker Recognition Evaluation*, 2010.

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

[14] M. H. DeGroot, *Optimal Statistical Decisions*. McGraw-Hill, 1970.

[15] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, "Discriminatively trained Probabilistic Linear Discriminant Analysis for speaker verification," in *Proceedings of ICASSP 2011*, pp. 4832–4835, 2011.

[16] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[17] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[20] S. Cumani and P. Laface, “Analysis of large–scale SVM training algorithms for language and speaker recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1585–1596, 2012.
[21] C. H. Teo, A. Smola, S. V. Vishwanathan, and Q. V. Le, “A scalable modular convex solver for regularized risk minimization,” in *Proceedings of KDD 2007*, pp. 727–736, 2007.
[22] C. H. Teo, A. Smola, S. V. Vishwanathan, and Q. V. Le, “Bundle methods for regularized risk minimization,” *J. Mach. Learn. Res.*, vol. 11, pp. 311–365, March 2010.
[23] V. Franc and S. Sonnenburg, “Optimized cutting plane algorithm for Support Vector Machines,” in *Proceedings of ICML 2008*, pp. 320–327, 2008.
[24] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions and Applications*. Springer-Verlag, 1985.
[25] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, “The entire regularization path for the Support Vector Machine,” *J. Mach. Learn. Res.*, vol. 5, pp. 1391–1415, December 2004.
[26] T. Joachims, “Making large–scale Support Vector Machine learning practical,” in *Advances in Kernel Methods – Support Vector Learning*, pp. 169–184, MIT–Press, 1999.
[27] A. Hatch, S. Kajarekar, and A. Stolcke, “Within-class covariance normalization for SVM-based speaker recognition,” in *Proceedings of ICSLP 2006*, pp. 1471–1474, 2006.
[28] M. Senoussaoui, P. Kenny, N. Brümmer, E. de Villiers, and P. Dumouchel, “Mixture of PLDA models in i–vector space for gender–independent speaker recognition,” in *Proc. of Interspeech 2011*, pp. 25–28, 2011.

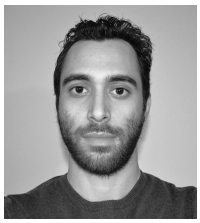
Sandro Cumani received the M.S. degree in Computer Engineering from the Politecnico di Torino, Torino, Italy, in 2008, and the Ph.D. degree in Computer and System Engineering of Politecnico di Torino in 2011. He is currently also with Brno University of Technology, Czech Republic. His current research interests include machine learning, speech processing and biometrics, in particular speaker and language recognition.

Niko Brümmer received B.Ing (1986), M.Ing (1988) and Ph.D.(2010) degrees in Electronic Engineering from Stellenbosch University, South Africa. He worked as researcher at DataFusion (later called Spescom DataVoice) and is currently chief scientist at AGNITIO. Most of his research for the last two decades has been applied to automatic speaker and language recognition and he has been participating in most of the NIST SRE and LRE evaluations in these technologies, from the year 2000 to the present. He was co-chair of Odyssey 2008: The Speaker and Language Recognition Workshop in Stellenbosch. His FoCal Toolkit is widely used for fusion and calibration in speaker and language recognition research. His interests include development of new algorithms for speaker and language recognition, as well as evaluation methodologies for these technologies. In both cases, his emphasis is on probabilistic modeling.

Lukáš Burget (Ing. [MS]. Brno University of Technology, 1999, Ph.D. Brno University of Technology, 2004) is assistant professor at Faculty of Information Technology, University of Technology, Brno, Czech Republic. He serves as scientific director of the Speech@FIT research group. From 2000 to 2002, he was a visiting researcher at OGI Portland, USA and from 2011 to 2012 he spent his sabbatical leave at SRI International, Menlo Park, USA. Lukas was invited to lead the Robust Speaker Recognition over Varying Channels team at the Johns Hopkins University CLSP summer workshop in 2008, and the team of BOSARIS workshop in 2010. His scientific interests are in the field of speech processing, namely acoustic modeling for speech, speaker and language recognition, including their software implementations. He has authored or co-authored more than 110 papers in journals and conferences. Dr. Burget is member of IEEE and ISCA.

Pietro Laface received the M.S. degree in Electronic Engineering from the Politecnico di Torino, Torino, Italy, in 1973.
Since 1988 it has been full Professor of Computer Science at the Dipartimento di Automatica e Informatica of Politecnico di Torino, where he leads the speech technology research group. He has published over 120 papers in the area of pattern recognition, artificial intelligence, and spoken language processing. His current research interests include all aspects of automatic speech recognition and its applications, in particular speaker and spoken language recognition.

Oldřich Plchot received Master degree in Computer Science and Engineering from Brno University of Technology, Czech Republic, in 2007, and he is pursuing the Ph.D. degree in Computer Science and Engineering at the same University. His current research interests include machine learning, data engineering, speech processing and biometrics, in particular speaker and language recognition.



Vasileios Vasilakakis Vasileios Vasilakakis obtained his MSc in Artificial Intelligence at University of Edinburgh, UK, 2009. He is currently pursuing a PhD in Forensic Speaker Verification at the Politecnico di Torino in Italy.
His research interest includes deep learning and machine learning techniques and their application in biometrics and forensics. Currently he is focusing on the use of deep belief networks for speaker verification.