DISCRIMINATIVELY TRAINED PROBABILISTIC LINEAR DISCRIMINANT ANALYSIS FOR SPEAKER VERIFICATION

Lukáš Burget[1], Oldřich Plchot[1], Sandro Cumani[2], Ondřej Glembek[1], Pavel Matějka[1], Niko Brümmer[3]

[1]Brno University of Technology, Czech Rep., {burget,iplchot,glembek,matejkap}@fit.vutbr.cz,
[2]Politecnico di Torino, Italy, sandro.cumani@polito.it, [3]AGNITIO, S. Africa, niko.brummer@gmail.com

ABSTRACT

Recently, i-vector extraction and Probabilistic Linear Discriminant Analysis (PLDA) have proven to provide state-of-the-art speaker verification performance. In this paper, the speaker verification score for a pair of i-vectors representing a trial is computed with a functional form derived from the successful PLDA generative model. In our case, however, parameters of this function are estimated based on a discriminative training criterion. We propose to use the objective function to directly address the task in speaker verification: discrimination between same-speaker and different-speaker trials. Compared with a baseline which uses a generatively trained PLDA model, discriminative training provides up to 40% relative improvement on the NIST SRE 2010 evaluation task.

Index Terms— Speaker verification, Discriminative training, Probabilistic Linear Discriminant Analysis

1. INTRODUCTION

In this paper, we show that discriminative training can be used to improve the performance of state-of-the-art speaker verification systems based on i-vector extraction and Probabilistic Linear Discriminant Analysis (PLDA). Recently, systems based on i-vectors [1, 2] extracted from cepstral features have provided superior performance in speaker verification. The so-called i-vector is an information-rich low-dimensional fixed length vector extracted from the feature sequence representing a speech segment (see section 2 for more details on i-vector extraction). A speaker verification score is then produced by comparing the two i-vectors corresponding to the segments in the verification trial. The function taking two i-vectors as an input and producing the corresponding verification score is typically designed to give a good approximation of the log-likelihood ratio between the "same-speaker" and "different-speaker" hypotheses. Typically, the function is also designed to produce a symmetric score (i.e. to produce output that is independent of which segment is enrollment and which is test — unlike traditional systems, which distinguish the two). In [1], good performance was reported when scores were computed as cosine distances between i-vectors normalized using within-class covariance normalization (WCCN). Best performance, however, is currently obtained with PLDA [2] — a generative model that models i-vector distributions allowing for direct evaluation of the desired log-likelihood ratio verification score (see section 3 for details on the specific form of PLDA used in our work).

In this paper, we propose to estimate verification scores using a *discriminative model* rather than a generative PLDA model. More specifically, the speaker verification score for a pair of i-vectors is computed using a function having the functional form derived from the PLDA generative model. The parameters of the function, however, are estimated using a discriminative training criterion. We use an objective function that directly addresses the speaker verification task, i.e. the discrimination between "same-speaker" and "different-speaker" trials. In other words, a binary classifier that takes a pair of i-vectors as an input, is trained to answer the question of whether or not the two i-vectors come from the same speaker. We show that the functional form derived from PLDA can be interpreted as a binary linear classifier in a nonlinearly expanded space of i-vector pairs. We have experimented with two discriminative linear classifiers, namely linear support vector machines (SVM) and logistic regression. The advantage of logistic regression is its probabilistic interpretation: the linear output of this classifier can be directly interpreted as the desired log-likelihood ratio verification score. On the NIST SRE 2010 evaluation task, we show that up to 40% relative improvement over the PLDA baseline can by obtained with such discriminatively trained models.

There has been previous work on discriminative training for speaker recognition, such as GMM-SVM [3]. This and similar approaches, however, do not directly address the objective of discriminating between same-speaker and different-speaker trials. Instead, SVMs are trained as discriminative models representing each target speaker. As a consequence, this approach cannot fully benefit from discriminative training, as there is a very limited number of positive examples (usually only one enrollment segment) available for training of each model. In contrast, in our approach, a model is trained using a large number of positive and negative examples, each of which is one of many possible same-speaker or different-speaker trials that can be constructed from the training segments.

The very same idea of discriminatively training a PLDA-like model for speaker verification was originally proposed in [4] and some initial work has been done in [5]. At that time, however, speaker factors extracted using Joint Factor Analysis (JFA) [6] were used as a suboptimal input for the classifier, and state-of-the-art performance would not have been achieved.

2. I-VECTORS

The i-vector approach has become state of the art in the speaker verification field [1]. The approach provides an elegant way of reducing large-dimensional input data to a small-dimensional feature vector while retaining most of the relevant information. The technique was originally inspired by the JFA framework [6]. The basic principle is that on some data, we train the i-vector extractor and then for

This work was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the Army Research Laboratory (ARL). All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI, or the U.S. Government. The work was also partly supported by the Grant Agency of Czech Republic project No. 102/08/0707, and Czech Ministry of Education project No. MSM0021630528.

each speech segment, we extract the i-vector as a low-dimensional fixed length representation of the segment. The main idea is that the speaker- and session-dependent supervectors of concatenated Gaussian mixture model (GMM) means can be modeled as

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

where $\mathbf{m}$ is the Universal Backgroung Model (UBM) GMM mean supervector, $\mathbf{T}$ is a matrix of bases spanning the subspace covering the important variability (both speaker- and session-specific) in the supervector space, and $\mathbf{x}$ is a standard-normally distributed latent variable. For each observation sequence representing a segment, our i-vector $\boldsymbol{\phi}$ is the MAP point estimate of the latent variable $\mathbf{x}$.

3. PLDA

3.1. Two covariance model

To facilitate comparison of i-vectors in a verification trial, we model the distribution of i-vectors using a Probabilistic LDA model [7, 2]. We first consider only a special form of PLDA, a *two-covariance model*, in which speaker and inter-session variability are modeled using across-class and within-class full covariance matrices $\boldsymbol{\Sigma}_{ac}$ and $\boldsymbol{\Sigma}_{wc}$. The two-covariance model is a generative linear-Gaussian model, where latent vectors $\mathbf{y}$ representing speakers (or more generally classes) are assumed to be distributed according to prior distribution

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}_{ac}). \tag{2}$$

For a given speaker represented by a vector $\hat{\mathbf{y}}$, the distribution of i-vectors is assumed to be

$$p(\boldsymbol{\phi}|\hat{\mathbf{y}}) = \mathcal{N}(\boldsymbol{\phi}; \hat{\mathbf{y}}, \boldsymbol{\Sigma}_{wc}). \tag{3}$$

The ML estimates of the model parameters, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}_{ac}$, and $\boldsymbol{\Sigma}_{wc}$, can be obtained using an EM algorithm as in [2]. The training i-vectors come from a database comprising recordings of many speakers (to capture across-class variability), each recorded in several sessions (to capture within-class variability).

In the more general case, the speaker and/or inter-session variability can be modeled using subspaces [1]. For example, in our baseline system, speaker variability is not modeled using a full covariance matrix. Instead a low rank across-class covariance matrix is modeled as $\boldsymbol{\Sigma}_{ac} = \mathbf{V}^T\mathbf{V}$, which limits speaker variability to live in a subspace spanned by the columns of the reduced rank matrix V.

3.2. Evaluation of verification score

Consider the process of generating two i-vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ forming a trial. In the case of a same-speaker trial, a single vector $\hat{\mathbf{y}}$ representing a speaker is generated from the prior $p(\mathbf{y})$, for which both $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ are generated from $p(\boldsymbol{\phi}|\hat{\mathbf{y}})$. For a different-speaker trial, two latent vectors representing two different speakers are independently generated from $p(\mathbf{y})$. For each latent vector, one of the i-vectors $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ is generated. Given a trial, we want to test two hypotheses: $\mathcal{H}_d$ that the trial is a different-speaker trial and $\mathcal{H}_s$ that the trial is a same-speaker trial. The speaker verification score can now be calculated as a log-likelihood ratio between the two hypotheses $\mathcal{H}_s$ and $\mathcal{H}_d$ as

$$\begin{aligned} s &= \log \frac{p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathcal{H}_s)}{p(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathcal{H}_d)} && (4) \\ &= \log \frac{\int p(\boldsymbol{\phi}_1|\mathbf{y})p(\boldsymbol{\phi}_2|\mathbf{y})p(\mathbf{y})d\mathbf{y}}{p(\boldsymbol{\phi}_1)p(\boldsymbol{\phi}_2)}, && (5) \end{aligned}$$

where in the numerator we integrate over the distribution of speaker vectors and, for each possible speaker, the likelihood of producing both i-vectors from the speaker is calculated. In the denominator, we simply multiply the marginal likelihoods $p(\boldsymbol{\phi}) = \int p(\boldsymbol{\phi}|\mathbf{y})p(\mathbf{y})d\mathbf{y}$. The integrals, which can be interpreted as convolutions of Gaussians, can be evaluated analytically giving

$$\begin{aligned} s &= \log \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{tot} & \boldsymbol{\Sigma}_{ac} \\ \boldsymbol{\Sigma}_{ac} & \boldsymbol{\Sigma}_{tot} \end{bmatrix} \right) \\ &\quad - \log \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{tot} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{tot} \end{bmatrix} \right), \end{aligned} \tag{6}$$

where the total covariance matrix is given as $\boldsymbol{\Sigma}_{tot} = \boldsymbol{\Sigma}_{ac} + \boldsymbol{\Sigma}_{wc}$. By expanding the log of Gaussian distributions and simplifying the final expression, we obtain

$$\begin{aligned} s &= \boldsymbol{\phi}_1^T\boldsymbol{\Lambda}\boldsymbol{\phi}_2 + \boldsymbol{\phi}_2^T\boldsymbol{\Lambda}\boldsymbol{\phi}_1 + \boldsymbol{\phi}_1^T\boldsymbol{\Gamma}\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2^T\boldsymbol{\Gamma}\boldsymbol{\phi}_2 \\ &\quad + (\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2)^T\mathbf{c} + k, \end{aligned} \tag{7}$$

where

$$\begin{aligned} \boldsymbol{\Gamma} &= -\frac{1}{4}(\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1} - \frac{1}{4}\boldsymbol{\Sigma}_{wc}^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_{tot}^{-1} \\ \boldsymbol{\Lambda} &= -\frac{1}{4}(\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1} + \frac{1}{4}\boldsymbol{\Sigma}_{wc}^{-1} \\ \mathbf{c} &= ((\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1} - \boldsymbol{\Sigma}_{tot}^{-1})\boldsymbol{\mu} \\ k &= \log|\boldsymbol{\Sigma}_{tot}| - \frac{1}{2}\log|\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac}| - \frac{1}{2}\log|\boldsymbol{\Sigma}_{wc}| \\ &\quad + \boldsymbol{\mu}^T(\boldsymbol{\Sigma}_{tot}^{-1} - (\boldsymbol{\Sigma}_{wc} + 2\boldsymbol{\Sigma}_{ac})^{-1})\boldsymbol{\mu}. \end{aligned} \tag{8}$$

We recall that the computation of a 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 $\mathrm{vec}(\cdot)$ stacks the columns of a matrix into a vector. Therefore, the log-likelihood ratio score can be written as a dot product of a vector of weights $\mathbf{w}^T$, and an expanded vector $\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ representing a trial:

$$\begin{aligned} s &= \mathbf{w}^T\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \\ &= \begin{bmatrix} \mathrm{vec}(\boldsymbol{\Lambda}) \\ \mathrm{vec}(\boldsymbol{\Gamma}) \\ \mathbf{c} \\ k \end{bmatrix}^T \begin{bmatrix} \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_2^T + \boldsymbol{\phi}_2\boldsymbol{\phi}_1^T) \\ \mathrm{vec}(\boldsymbol{\phi}_1\boldsymbol{\phi}_1^T + \boldsymbol{\phi}_2\boldsymbol{\phi}_2^T) \\ \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 \\ 1 \end{bmatrix}. \end{aligned} \tag{9}$$

Hence, we have obtained a generative generalized linear classifier [8], where the probability for a same-speaker trial can be computed from the log-likelihood ratio score using the sigmoid activation function as

$$p(\mathcal{H}_s|\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \sigma(s) = (1 + \exp(-s))^{-1}. \tag{10}$$

Here, we have assumed equal priors for both hypotheses. To allow for different priors, we can simply adjust the constant k in the vector of weights by adding $\mathrm{logit}(p(\mathcal{H}_s))$.

4. DISCRIMINATIVE CLASSIFIERS

In this section, we describe how we train the weights $\mathbf{w}$ directly, in order to discriminate between same-speaker and different-speaker trials, without having to explicitly model the distributions of i-vectors. To represent a trial, we keep the same expansion $\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ as defined in (9). Hence, we reuse the functional form for computing verification scores that provided excellent results with generative PLDA. We consider two standard discriminative linear classifiers, namely logistic regression and SVMs.

4.1. Objective functions

The set of training examples, which we continue referring to as training trials, comprises both different-speaker and same-speaker trials. Let us use the coding scheme $t \in \{-1, 1\}$ to represent labels for the different-speaker, and same-speaker trials, respectively. Assigning each trial a log-likelihood ratio s and the correct label t, the log probability of recognizing the trial correctly can be expressed as

$$\log p(t|\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = -\log(1 + \exp(-st)). \tag{11}$$

This is easy to see from equation (10) and recalling that $\sigma(-s) = 1 - \sigma(s)$. In the case of logistic regression, the objective function to maximize is the log probability of correctly classifying all training examples, i.e. the sum of expressions (11) evaluated for all training trials. Equivalently, this can be expressed by minimizing the cross-entropy error function, which is a sum over all training trials

$$E(\mathbf{w}) = \sum_{n=1}^{N} \alpha_n E_{LR}(t_n s_n) + \frac{\lambda}{2}\|\mathbf{w}\|^2, \tag{12}$$

where the logistic regression loss function

$$E_{LR}(ts) = \log(1 + \exp(-ts)) \tag{13}$$

is simply the negative log probability (11) of correctly recognizing a trial. We have also added the regularization term $\frac{\lambda}{2}\|\mathbf{w}\|^2$, where λ is a constant controlling the tradeoff between the error function and the regularizer. The coefficients α_n allow us to weight individual trials. Specifically, we use them to assign different weights to same-speaker and different-speaker trials. This allows us to select a particular operating point, around which we want to optimize the performance of our system without relying on the proportion of same- and different-speaker trials in the training set. The advantage of using the cross-entropy objective for training is that it reflects performance of the system over a wide range of operating points (around the selected one). For this reason, a similar function was also proposed as a performance measure for the speaker verification task [9]. Another advantage of using the logistic regression classifier is its probabilistic nature: It trains the weights so that the score $s = \mathbf{w}^T\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ can be interpreted as the log-likelihood ratio between hypotheses $\mathcal{H}_s$ and $\mathcal{H}_d$.

Taking (12) and replacing $E_{LR}(ts)$ with hinge loss function

$$E_{SV}(ts) = \max(0, 1 - ts), \tag{14}$$

we obtain an SVM, which is a classifier traditionally understood to maximize the margin separating class samples. Alternatively, one can see the hinge loss function as a piecewise approximation to the logistic regression loss function. Therefore, one can assume that the score $s = \mathbf{w}^T\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ obtained from an SVM classifier will still be a reasonable approximation to the log-likelihood ratio (after a linear calibration).

4.2. Gradient evaluation

In order to numerically optimize the parameters $\mathbf{w}$ of the classifier, we want to evaluate the gradient of the error function

$$\nabla E(w) = \sum_{n=1}^{N} \alpha_n \frac{\partial E(t_n s_n)}{\partial s_n}\frac{\partial s_n}{\partial \mathbf{w}} + \lambda\mathbf{w}, \tag{15}$$

where the derivation of the loss function $E(t_n s_n)$, w.r.t. score s_n, depends on the particular choice of the loss function. For the logistic regression loss function, it is defined as

$$\frac{\partial E_{LR}(ts)}{\partial s} = -t\sigma(-ts) \tag{16}$$

while for the hinge loss function it becomes

$$\frac{\partial E_{SV}(ts)}{\partial s} = \begin{cases} 0 & \text{if } \; ts \geq 1 \\ -t & \text{otherwise.} \end{cases} \tag{17}$$

Finally, the derivation of the score w.r.t. the classifier parameters just gives the expanded trial vector

$$\frac{\partial s}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}\mathbf{w}^T\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2). \tag{18}$$

4.3. Efficient score and gradient evaluation

Given a trained classifier, we can obtain a verification score for a trial by forming the expanded vector $\boldsymbol{\varphi}(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ and computing the dot product (9). However, as we have already seen, the same score can be obtained using the two original i-vectors $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2$ and using the formula (7), which is both memory and computationally efficient. Now, consider two sets of i-vectors stored as columns of the matrices $\boldsymbol{\Phi}_e$ and $\boldsymbol{\Phi}_t$. For illustration, let us call these sets enrollment and test trials, although they play symmetrical roles in our scoring scheme. We can efficiently score each enrollment trial against each test trial and obtain the full matrix of scores as

$$\begin{aligned} \mathbf{S} \;=\; & 2\boldsymbol{\Phi}_e^T\boldsymbol{\Lambda}\boldsymbol{\Phi}_t \\ & +((\boldsymbol{\Phi}_e^T\boldsymbol{\Gamma}) \circ \boldsymbol{\Phi}_e^T)\mathbf{1}\mathbf{1}^T + \mathbf{1}\mathbf{1}^T(\boldsymbol{\Phi}_t \circ (\boldsymbol{\Gamma}\boldsymbol{\Phi}_t)) \\ & +\boldsymbol{\Phi}_e^T\mathbf{c}\mathbf{1}^T + \mathbf{1}\mathbf{c}^T\boldsymbol{\Phi}_t + k\mathbf{1}\mathbf{1}^T, \end{aligned} \tag{19}$$

where $\circ$ denotes the Hadamard, or "entrywise" product. Similarly, the naïve way of evaluating the gradient would be to explicitly expand every training trial and then to apply equations (15) to (18). However, again taking into account the functional form for computing scores (7), the gradient can be evaluated much more efficiently without any need for explicit trial expansion. Let all the i-vectors, which we have available for training, be stored in columns of a matrix $\boldsymbol{\Phi}$. Now consider forming a training trial using every possible pair of i-vectors from the matrix. Let s_{ij} be the score for the trial formed by the i-th and j-th columns of $\boldsymbol{\Phi}$ calculated using the parameters $\mathbf{w}$ for which we wish to evaluate the gradient. Let t_{ij} and α_{ij} be the corresponding label and trial weight, respectively. Further, let d_{ij} be the corresponding derivation of loss function $E(t_{ij}s_{ij})$ w.r.t. the score s_{ij} given in (16) or (17) depending on the loss function used. The gradient can now be efficiently evaluated as

$$\nabla E(\mathbf{w}) = \begin{bmatrix} \nabla_\Lambda L \\ \nabla_\Gamma L \\ \nabla_c L \\ \nabla_k L \end{bmatrix} = \begin{bmatrix} 2 \cdot \mathrm{vec}\left(\boldsymbol{\Phi}\mathbf{G}\boldsymbol{\Phi}^T\right) \\ 2 \cdot \mathrm{vec}\left(\boldsymbol{\Phi}[\boldsymbol{\Phi}^T \circ (\mathbf{G}\mathbf{1}\mathbf{1}^T)]\right) \\ 2 \cdot \mathbf{1}^T[\boldsymbol{\Phi}^T \circ (\mathbf{G}\mathbf{1}\mathbf{1}^T)] \\ \mathbf{1}^T\mathbf{G}\mathbf{1} \end{bmatrix} + \lambda\mathbf{w} \tag{20}$$

where elements of matrix $\mathbf{G}$ are $g_{ij} = d_{ij} \cdot \alpha_{ij}$.

5. EXPERIMENTS

The i-vector extractor and the baseline PLDA system is taken from the ABC system submitted to NIST SRE 2010 evaluation [10]. The i-vector extractor uses 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. All PLDA systems and discriminative classifiers are trained using 400 dimensional i-vectors extracted from 21663 segments from 1384 female speakers and 16969 segments from 1051 male speakers from NIST SRE 2004, NIST SRE 2005, NIST SRE 2006, Switchboard II Phases 2 and 3, and Switchboard Cellular Parts 1 and 2. Table 1 presents results for the extended condition 5 (tel-tel)

	Female Set			Male Set			Pooled		
System	minDCF	oldDCF	EER	minDCF	oldDCF	EER	minDCF	oldDCF	EER
PLDA	0.40	0.15	3.57	0.42	0.13	2.86	0.41	0.14	3.23
LR	0.40	0.12	2.94	0.39	0.10	2.22	0.40	0.11	2.62
SVM	0.39	0.11	2.35	0.31	0.08	1.55	0.37	0.10	1.94
HT-PLDA	0.34	0.11	2.22	0.33	0.08	1.47	0.34	0.10	1.88

Table 1. Normalized newDCF, oldDCF and EER for the extended condition 5 (tel-tel) from the NIST SRE 2010 evaluation.

from NIST SRE 2010 evaluation. The reported numbers are Equal Error Rate (EER) and normalized minimum Decision Cost Functions for the two operating points as defined by NIST for the SRE 2008 (oldDCF) and SRE 2010 (newDCF) evaluations [11].

The system denoted as PLDA, which serves as our baseline, is based on a generatively trained PLDA model with a 90-dimensional speaker variability subspace [10]. On telephone data, this configuration was found to give the best newDCF, which was the primary performance measure in the NIST SRE 2010 evaluation, which focused on low false alarm rates. As a tradeoff, the system gives somewhat poorer performance at the oldDCF and EER.

The system denoted as LR is the discriminative linear classifier, where parameters were initialized from the baseline system using (8) and retrained to optimize the logistic regression objective function. We have used the conjugate gradient trust region method [12] as implemented in [13] to numerically optimize the parameters. No regularization was used in this case. Significant improvements compared to the baseline can be observed, especially at oldDCF and EER.

Even larger improvements were observed for the SVM-based classifier, where 10%, 30% and 40% relative improvements over the baseline were obtained for newDCF, oldDCF and EER respectively. The improvements over the LR system can probably be attributed mainly to the presence of the regularization term. Often, SVM classifiers are trained using a solver to the dual problem, where a Gram matrix needs to be evaluated. The Gram matrix is a matrix comprising dot products between every pair of training examples, which are the trials in our case. Since we decided to construct a training trial for every pair of i-vectors, the size of the Gram matrix would be unmanageably large (the number of training i-vectors to the 4th power). Therefore, we train a linear SVM by again solving the primal problem using a solver [14], which makes use of the efficient evaluation of gradient. To make SVM regularization effective, we have found that it is necessary to first normalize input i-vectors using within-class covariance normalization (WCCN) [1], i.e. to normalize i-vectors to have identity within-class covariance matrix. More details on the SVM-based system described in this paper can be found in our parallel paper [15].

Finally, for comparison, we also include results with Heavy-tailed PLDA (HT-PLDA) [2], which are so far the best results we have obtained with the same set of training and test i-vectors. In heavy-tailed PLDA, speaker and intersession variability are modeled using Student's t, rather than Gaussian distributions. In our system, the dimensionality of i-vectors was first reduced from 400 to 120 and the final vectors were modeled with full-rank speaker and intersession subspaces. Nevertheless, the price paid for the excellent results obtained with heavy-tailed PLDA is the very computationally demanding score evaluation. As we can see, competitive results can be obtained with our discriminatively trained models, for which the score evaluation is several orders of magnitude faster.

6. CONCLUSIONS

Recent advances in speaker verification build on i-vector extraction and Probabilistic Linear Discriminant Analysis (PLDA). In this paper, we have proposed to use a PLDA-like functional for evaluating the speaker verification score for a pair of i-vectors representing a trial. However, estimation of the function parameters is based on a discriminative rather than a generative training criterion. We have shown the benefit of using the objective function to directly address the task in speaker verification: discrimination between same-speaker and different-speaker trials. On the NIST SRE 2010 evaluation task, our results show a significant (up to 40%) relative improvement from this approach, compared to a baseline that uses a generatively trained PLDA model.

In future work, we would like to test our method on additional conditions beyond the telephone speech, and to develop techniques for adapting the trained system to be able to cope with new channel conditions. Various methods for regularizing logistic regression training are also worth investigating. We would also like to experiment with models based on more general forms of the PLDA model. Functional forms for verification scores derived from PLDA with low-rank speaker or channel subspaces would allow us to control the number of trainable parameters. Another interesting alternative would be a functional form that would more closely simulate the heavy-tailed PLDA generative model [2], which is currently providing better performance than PLDA based on Gaussian distributions.

7. REFERENCES

[1] N. Dehak, P. Kenny, et al., "Front–end factor analysis for speaker verification," in *IEEE Trans. on Audio, Speech and Lang. Process.*, 2010.

[2] P. Kenny, "Bayesian speaker verification with heavy–tailed priors," keynote presentation, Proc. of Odyssey 2010, June 2010.

[3] 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," May 2006, vol. 1, pp. I –I.

[4] N. Brümmer, "A farewell to SVM: Bayes factor speaker detection in supervector space," http://sites.google.com/site/nikobrummer/.

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

[6] P. Kenny et al., "Joint factor analysis versus eigenchannes in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

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

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*, chapter 4.2, Springer, 2006.

[9] N. Brümmer and J. A. du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

[10] N. Brummer, L. Burget, P. Kenny, et al., "ABC system description for NIST SRE 2010," in *Proc. NIST 2010 Speaker Recognition Evaluation*.

[11] NIST, "The NIST year 2008 and 2010 speaker recognition evaluation plans," http://www.itl.nist.gov/iad/mig/tests/sre.

[12] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer, August 2000.

[13] E. de Villiers and N. Brümmer, "BOSARIS toolkit," https://sites.google.com/site/bosaristoolkit/.

[14] C.H. Teo, A. Smola, et al., "A scalable modular convex solver for regularized risk minimization," in *Proc. of KDD*, 2007, pp. 727–736.

[15] S. Cumani, N. Brummer L. Burget, , and P. Laface, "Fast discriminative speaker verification in the i-vector space," submitted to Proc. of ICASSP 2011, Prague.