Odyssey 2010
The Speaker and Language Recognition Workshop
28 June – 1 July 2010, Brno, Czech Republic

Weighted Nuisance Attribute Projection*

W. M. Campbell

MIT Lincoln Laboratory, Lexington, MA
wcampbell@ll.mit.edu

Abstract

Nuisance attribute projection (NAP) has become a common method for compensation of channel effects, session variation, speaker variation, and general mismatch in speaker recognition. NAP uses an orthogonal projection to remove a nuisance subspace from a larger expansion space that contains the speaker information. Training the NAP subspace is based on optimizing pairwise distances to reduce intraspeaker variability and retain interspeaker variability. In this paper, we introduce a novel form of NAP called weighted NAP (WNAP) which significantly extends the current methodology. For WNAP, we propose a training criterion that incorporates two critical extensions to NAP—variable metrics and instance-weighted training. Both an eigenvector and iterative method are proposed for solving the resulting optimization problem. The effectiveness of WNAP is shown on a NIST speaker recognition evaluation task where error rates are reduced by over 20%.

Index Terms: speaker recognition, channel compensation

1. Introduction

A problem of primary importance in speaker recognition is compensation for intraspeaker recording variation. Sources of variation can be—microphone types, communication channels, source encoding, noise type and levels, intrinsic speaker variability, etc. Many of the common methods for speaker variation compensation have targeted a particular type of variability. For instance, cepstral mean subtraction (CMS) attempts to eliminate variability from linear time invariant filters with low group delay. CMS is quite effective at reducing variation due to spectral tilt and shaping that is common with varying microphone types.

Rather than try to model the *physics* of all different types of intraspeaker variation, it is possible to take a data-driven approach. In this case, with the wide availability of different large multisession corpora such as the LDC Switchboard and Mixer corpora, we can observe large amounts of intraspeaker variation through multisession recording. Although not explicit controlled, the collection protocols of the various corpora ensure variation due to many of the factors stated above.

The basic approach with NAP is to take advantage of large corpora intraspeaker variation to train a model that discriminatively reduces the nuisance component. In SVM speaker recognition [1], directions in expansion space correspond to classifiers, so it is straightforward to view channel effects as nuisance directions that should be removed. In early NAP experiments, both channel information (cell, carbon button, and electret) and session information were used to train the NAP projection [2]. Later work showed that session variation was sufficient [3].

A significant amount of work has been performed in data-driven methods since NAP and factor analysis methods [4] for speaker recognition were first proposed. Most of these techniques have focused on new methods for compensation or model construction including WCCN [5], joint factor analysis [6], etc. Less work has been performed on alternate criteria for optimizing the hyperparameters for these methods [7].

In this paper, we propose an alternate criterion and extension to NAP—Weighted NAP (WNAP). The main motivation for this extension is to address new aspects of metrics induced by inner product discriminant functions (IPDFs) [8]. First, WNAP addresses variable metrics where the metric is not fixed across all utterances. Second, WNAP incorporates a variable weighting across utterances. This feature allows WNAP to be trained to address issues such priors in the data set (e.g., male/female distribution), confidence in the SVM expansion due to speech duration, etc.

The outline of the paper is as follows. In Section 2, we review the IPDF framework. In Section 3, we review NAP and the associated training criterion. Sections 4, 5, 6 provide the main discussion of WNAP and various solutions. Section 7 provides pseudo-code for the various methods. Section 8 provides experiments demonstrating the new WNAP method and corresponding improvements in performance.

2. Inner Product Discriminant Functions

Inner product discriminant functions (IPDFs) [8] are a unified description of early work in inner-product based speaker recognition [1, 3, 9], data-driven subspace compensation methods such as NAP [3, 2] and factor analysis [6], and recent work in linear GMM scoring [10]. Although these methods have very distinct motivations and derivations, the resulting operations have very similar mathematical structure for both the comparison function (or inner product) and the compensation.

Before describing IPDFs, we introduce some notation. For a sequence of feature vectors from a speaker i, we adapt a GMM UBM by using standard relevance MAP [11] on the means and an ML estimate of the mixture weights. The adaptation yields new parameters which we stack into a parameter vector, $\mathbf{a}_i$, where

$$\mathbf{a}_i = \begin{bmatrix} \lambda_{i,1} & \cdots & \lambda_{i,N_m} & \mathbf{m}_{i,1}^t & \cdots & \mathbf{m}_{1,N_m}^t \end{bmatrix}^t \quad (1)$$

where $\lambda_{i,j}$ are the mixture weights, $\mathbf{m}_{i,j}$ are the means, and N_m is the number of mixtures.

To compare two speakers i and j, we use an inner product, but do not require the Mercer condition used in standard SVM speaker recognition. The IPDF in equation form is

$$C(\mathbf{a}_i, \mathbf{a}_j) = (L_i \mathbf{a}_i)^t D_{i,j}^2 (L_j \mathbf{a}_j) \quad (2)$$

*This work was sponsored by the Federal Bureau of Investigation under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

where L_i, L_j are linear transforms and are potentially dependent on mixture weights. The matrix $D_{i,j}$ is positive definite, usually diagonal, and potentially dependent on the mixture weights. Most of the standard linear scoring methods in speaker recognition plus several new ones can be expressed in the IPDF framework (2) by various forms of $D_{i,j}$ and taking subsets of $\mathbf{a}_i$ and $\mathbf{a}_j$. Most compensation methods can be expressed as linear projections or regularizations of projections. For more details, see [8].

We use a comparison function from the IPDF framework based on approximations to the KL divergence between two GMMs [3, 8], C_{GM}, given by

$$C_{GM}(\mathbf{a}_i, \mathbf{a}_j) = (\mathbf{m}_i - \mathbf{m})^t(\boldsymbol{\lambda}_i^{1/2} \otimes I_n)\Sigma^{-1}(\boldsymbol{\lambda}_j^{1/2} \otimes I_n)(\mathbf{m}_j - \mathbf{m}). \quad (3)$$

In equation (3), $\mathbf{m}$ is the vector of stacked UBM means, Σ is the block diagonal matrix of UBM covariances, $\otimes$ is the Kronecker product, I_n is the identity matrix of size n, and $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_j$ are diagonal matrices of mixture weights from (1).

In cases where a comparison function corresponds to a metric on the space, a unique corresponding distance can be defined as

$$d(\mathbf{a}_i, \mathbf{a}_j)^2 = C(\mathbf{a}_i, \mathbf{a}_i) - 2C(\mathbf{a}_i, \mathbf{a}_j) + C(\mathbf{a}_j, \mathbf{a}_j). \quad (4)$$

In general, we would like to be able to optimize compensation method for an arbitrary distance measure—we examine this process in the next few sections.

3. Nuisance Attribute Projection

As mentioned in the introduction, nuisance attribute projection (NAP) can be motivated as a method of removing nuisance directions from the SVM expansion space. If these directions are not removed, then utterances can be similar just based on the fact that they have similar nuisance content—for example, they were recorded from the same channel. To remove the nuisance, we use an orthogonal projection.

Before defining the NAP projection, we introduce some notation. We define an orthogonal projection with respect to a metric, $P_{U,D}$, where D and U are full rank matrices as

$$P_{U,D} = U(U^tD^2U)^{-1}U^tD^2 \quad (5)$$

where DU is a linearly independent set, and the metric is $\|x - y\|_D = \|Dx - Dy\|_2$. The process of projection, e.g. $y = P_{U,D}b$, is equivalent to solving the least-squares problem, $\hat{x} = \text{argmin}_x \|Ux - b\|_D$ and letting $y = U\hat{x}$. For convenience, we also define the projection onto the orthogonal complement of U, $U^\perp$, as $Q_{U,D} = P_{U^\perp,D} = I - P_{U,D}$.

If U is the nuisance subspace, NAP can be concisely represented as $Q_{U,D}$. For a set of training vectors, $\{\mathbf{z}_i\}$, the criterion for training NAP is

$$\min_U \sum_{i,j} W_{i,j}\|Q_{U,D}\mathbf{z}_i - Q_{U,D}\mathbf{z}_j\|_D^2. \quad (6)$$

Typical weights, $W_{i,j}$, used are $W_{i,j} = 1$ if $\mathbf{z}_i$ and $\mathbf{z}_j$ are from the same speaker and $W_{i,j} = 0$, otherwise. The NAP training criterion can be shown to be equivalent to an eigenvector problem [2].

4. Weighted Nuisance Attribute Projection

Although NAP is a powerful framework for compensation, there are potential drawbacks when it is applied in the IPDF framework. First, since NAP relies on pairwise comparison (6), it is not possible to apply metrics that are dependent on the utterance; e.g., to use a norm dependent on the mixture weights which arises naturally from the C_{GM} function. A second reason to consider extensions to NAP is to incorporate novel utterance dependent weightings into the optimization process. In the original framework in (6), since $W_{i,j}$ is dependent on a pair of instances, it is difficult to assign weights that are not 0 or 1.

To address these issues, we introduce Weighted NAP (WNAP). For WNAP, we assume a general projection onto $U^\perp$ of the form Q_{U,D_i}. We also introduce a training criterion based upon observations from earlier work [3]. Instead of considering pairwise comparison of instances, we assume that for every speaker (in general, every class) we can estimate a "nuisance free" vector $\bar{\mathbf{z}}$ from which deltas can be calculated. We will then base our criterion on approximating these deltas.

More specifically, suppose we have a training set, $\{\mathbf{z}_{s,i}\}$ labeled by speaker, s, and instance, i. For each s, we have a nuisance-free vector, $\bar{\mathbf{z}}_s$. For WNAP training, we propose the following optimization problem,

$$\min_U \sum_s \sum_i W_{s,i}\|P_{U,D_{s,i}}\delta_{s,i} - \delta_{s,i}\|_{D_{s,i}}^2 \quad (7)$$

where $\delta_{s,i} = \mathbf{z}_{s,i} - \bar{\mathbf{z}}_s$. The WNAP training criterion (7) incorporates both our goals of using a variable metric and an utterance dependent weighting. Also, the training criterion attempts to find a subspace U that best approximates the nuisance $\delta_{s,i}$ as in prior work [3].

5. Optimizing the WNAP Criterion

As a first step in optimizing the WNAP criterion, we consider a slightly more general version of the problem in (7) which will be useful in later sections. We relabel the data with one index i in (7). Also, rather than working with the projection, P_{U,D_i}, we work with coordinates, $\mathbf{x}_i$, in the U subspace. The problem we now consider is

$$\min_{U,\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2. \quad (8)$$

We split the variables into two parts, the subspace and the coordinate optimization, and minimize over these separately,

$$\begin{aligned} &\min_{U,\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \\ &= \min_U \min_{\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \quad (9) \\ &= \min_U \sum_{i=1}^N W_i \min_{\mathbf{x}_i} \|U\mathbf{x}_i - \delta_i\|_{D_i}^2. \end{aligned}$$

Note that in (9), the cascade of two min terms means hold U constant and then minimize over the $\mathbf{x}_1, \cdots, \mathbf{x}_N$. It is straightforward to prove that the cascade minimization has the same value as the simultaneous minimization. In (9), we also use the fact that the sum becomes a separable optimization problem when U is fixed. That is, we can minimize each term in the sum separately over $\mathbf{x}_i$.

For a fixed U, the solution to the least-squares problem,

$$\mathbf{x}_i^* = \underset{\mathbf{x}_i}{\operatorname{argmin}} \|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \tag{10}$$

is just the projection onto the subspace using the D_i metric. I.e., assuming that U is full rank, we have

$$\begin{aligned} U\mathbf{x}_i^* &= P_{U,D_i}\delta_i \\ &= U(U^tD_i^2U)^{-1}U^tD_i^2\delta_i. \end{aligned} \tag{11}$$

We can now substitute (11) back into the original minimization problem to obtain,

$$\begin{aligned} &\min_{U,\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \\ &= \min_U \sum_{i=1}^N W_i\|P_{U,D_i}\delta_i - \delta_i\|_{D_i}^2 \\ &= \min_U \sum_{i=1}^N W_i\|Q_{U,D_i}\delta_i\|_{D_i}^2 \\ &= \min_U \sum_{i=1}^N \|Q_{U,D_i}\hat{\delta}_i\|_{D_i}^2. \end{aligned} \tag{12}$$

In (12), we incorporated the W_i into the δ_i by letting,

$$\hat{\delta}_i = \sqrt{W_i}\delta_i. \tag{13}$$

Note that in (12), we have shown equivalence with our original NAP criterion (7).

Since the least squares problem produced an orthonormal projection onto the subspace, we can rewrite (12) as

$$\begin{aligned} &\min_{U,\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \\ &= \min_U \sum_{i=1}^N \|\hat{\delta}_i\|_{D_i}^2 - \|P_{U,D_i}\hat{\delta}_i\|_{D_i}^2 \\ &= \max_U \sum_{i=1}^N \|P_{U,D_i}\hat{\delta}_i\|_{D_i}^2 \end{aligned} \tag{14}$$

The resulting optimization problem in (14) has a satisfying qualitative goal—find the subspace U that has the most "nuisance" energy (norm squared) when we project the weighted deltas onto it.

The solution to (14) is difficult because of the variable metric induced by the D_i; we'll address this later in Section 6. Therefore, for the remainder of this section, we assume D_i is a fixed matrix D. Note that our optimization problem still incorporates the variable weighting W_i, so we have, at least, a restricted solution to our original WNAP problem.

We rewrite the norm in (14) in terms of the trace, $\operatorname{tr}(\cdot)$, and then use the assumption that $D_i = D$ is constant to obtain

$$\begin{aligned} &\max_U \sum_{i=1}^N \|P_{U,D_i}\hat{\delta}_i\|_{D_i}^2 \\ &= \max_U \sum_{i=1}^N \operatorname{tr}\left[\left(D_iP_{U,D_i}\hat{\delta}_i\right)\left(D_iP_{U,D_i}\hat{\delta}_i\right)^t\right] \\ &= \max_U \sum_{i=1}^N \operatorname{tr}\left[DP_{U,D}\hat{\delta}_i\hat{\delta}_i^tP_{U,D}^tD\right] \\ &= \max_U \operatorname{tr}\left[DP_{U,D}\left(\sum_{i=1}^N \hat{\delta}_i\hat{\delta}_i^t\right)P_{U,D}^tD\right] \\ &= \max_U \operatorname{tr}\left[DP_{U,D}RP_{U,D}^tD\right] \end{aligned} \tag{15}$$

where R is the correlation matrix, $R = \sum_{i=1}^N \hat{\delta}_i\hat{\delta}_i^t$.

Since we are interested only in the subspace, we want to limit the solutions to (15). An obvious assumption is that we have an orthonormal basis for the subspace—i.e., U is orthonormal wrt to D, $U^tD^2U = I$. Combining this assumption with (11) and (15) yields

$$\begin{aligned} &\max_{U,U^tD^2U=I} \operatorname{tr}\left[DP_{U,D}RP_{U,D}^tD\right] \\ &= \max_{U,U^tD^2U=I} \operatorname{tr}\left[DUU^tD^2RD^2UU^tD\right] \\ &= \max_{\hat{U},\hat{U}^t\hat{U}=I} \operatorname{tr}\left[\hat{U}\hat{U}^t\hat{R}\hat{U}\hat{U}^t\right] \\ &= \max_{\hat{U},\hat{U}^t\hat{U}=I} \operatorname{tr}\left[\hat{U}^t\hat{U}\hat{U}^t\hat{R}\hat{U}\right] \\ &= \max_{\hat{U},\hat{U}^t\hat{U}=I} \operatorname{tr}\left[\hat{U}^t\hat{R}\hat{U}\right] \end{aligned} \tag{16}$$

where we have substituted $\hat{U} = DU$, $\hat{R} = DRD$, and we have used the fact that $\operatorname{tr}(ABC) = \operatorname{tr}(CAB)$.

Assuming unique eigenvalues, a solution to (16) is the k eigenvectors belonging to the k largest eigenvalues of the matrix $\hat{R}$ where k is the rank (number of columns) of U; call this solution U_k. Note that the solution has a nice structure for varying k. If we want the solutions for any projection, $k_0 < k$, we just subset to the first k_0 columns of U_k (assuming that the columns are ordered by eigenvalue largest to smallest).

6. An Iterative Solution to WNAP

In the prior Section 5, we showed that for a fixed metric, $D_i = D$, and a variable weighting, W_i, that the WNAP solution can be solved via an eigenvalue problem. In the general case (for IPDFs), both W_i and D_i vary with the utterance.

Examining (15), we see that if D_i is variable, then the projection can not be factored out of the sum to obtain an eigenvector solution. Instead, we must go back to the alternate WNAP problem (9). We use the split variable version of the problem,

$$\begin{aligned} &\min_{U,\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \\ &= \min_U \min_{\mathbf{x}_1,\cdots,\mathbf{x}_N} \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \\ &= \min_{\mathbf{x}_1,\cdots,\mathbf{x}_N} \min_U \sum_{i=1}^N W_i\|U\mathbf{x}_i - \delta_i\|_{D_i}^2. \end{aligned} \tag{17}$$

The split variable expression (17) can be used to create an alternating minimization optimization where we alternately optimize U and then $\mathbf{x}_1, \cdots, \mathbf{x}_N$. The alternating minimization problem is similar to the type of solution method we would see in an EM type algorithm [12, 6, 13]. The solution of the alternating optimization has the same properties as EM—convergence to a local minimum (no guarantee of global optimality) [14].

For the alternating optimization problem (17), we know how to solve the case for fixed U and varying $\mathbf{x}_1, \cdots, \mathbf{x}_N$ from Section 5. The case for fixed $\{\mathbf{x}_i\}$ is distinct, and we consider it next.

For fixed $\mathbf{x}_1, \cdots, \mathbf{x}_N$, we break out the sum in equation (17) in terms of the rows of U which we will denote as the row vectors, U_1^t, U_2^t, etc., can be written as

$$\begin{aligned}
&\min_U \sum_{i=1}^N W_i \|U\mathbf{x}_i - \delta_i\|_{D_i}^2 \\
&= \min_U \sum_{i=1}^N \sum_j W_i D_{i,j}^2 \left(U_j^t \mathbf{x}_i - \delta_{i,j}\right)^2 \\
&= \min_U \sum_j \sum_{i=1}^N W_i D_{i,j}^2 \left(U_j^t \mathbf{x}_i - \delta_{i,j}\right)^2 \\
&= \sum_j \min_U \left[\sum_{i=1}^N W_i D_{i,j}^2 \left(U_j^t \mathbf{x}_i - \delta_{i,j}\right)^2\right]
\end{aligned} \tag{18}$$

where $\delta_{i,j}$ is the jth entry of δ_i. $D_{i,j}$ in this case is the jth diagonal entry of the matrix D_i. The problem in (18) is separable in that for each fixed j, we can optimize separately the sums,

$$\begin{aligned}
&\min_{U_j} \sum_{i=1}^N W_i D_{i,j}^2 \left(U_j^t \mathbf{x}_i - \delta_{i,j}\right)^2 \\
&= \min_{U_j} \sum_{i=1}^N \left(W_i^{1/2} D_{i,j} \mathbf{x}_i^t U_j - W_i^{1/2} D_{i,j} \delta_{i,j}\right)^2.
\end{aligned} \tag{19}$$

In many cases, the matrix D_i has a fixed part, $\bar{D}$ and a variable part, $\tilde{D}_i$, so that $D_i = \bar{D}\tilde{D}_i$. For the least squares problem in (19), we only need consider the variable part and the resulting normal equations are

$$\left(\sum_{i=1}^N W_i \tilde{D}_{i,j}^2 \mathbf{x}_i \mathbf{x}_i^t\right) U_j = \sum_{i=1}^N W_i \tilde{D}_{i,j}^2 \delta_{i,j} \mathbf{x}_i \tag{20}$$

The normal equations (20) can be solved, for example, using a Cholesky decomposition and back substitution [15].

7. Implementing WNAP Training

To simplify easy of implementation, we provide pseudocode that describes the implementation of WNAP. In Algorithm 1, the WNAP solution for a fixed metric is given from Section 5. Note that this process could also be implemented via kernel methods as in prior work [3, 16].

Our algorithm for iterative training is given in Algorithm 2. Note that, as in Section 6, $\tilde{D}_{k,j}$ refers to the jth diagonal entry of the matrix $\tilde{D}_k$. Also, we have introduced a regularization constant ϵ which can be set to a small number, e.g., $\epsilon = 0.001$, to eliminate ill-conditioning issues. Finally, we mention that Algorithm 2 is not optimized for computation; in many cases, $\tilde{D}_k$ will contain redundant entries and so many R_j will be the same. For example, for $C_{GM}(\cdot)$, the appropriate $\tilde{D}_k$ is $\boldsymbol{\lambda}_k^{1/2} \otimes I_n$ which has only N_m unique entries (N_m equals the number of mixture components).

Algorithm 1 WNAP subspace training algorithm for a fixed metric, D, with the eigenvector method

Input: Data set $\{\mathbf{z}_i\}$ of N training vectors, weights $\{W_i\}$, with speaker labels $\{l_i\}$, and the desired corank
Output: Nuisance subspace, U
for all s in unique speakers in $\{l_i\}$ **do**
  Find $\bar{\mathbf{z}}_s$
  for all j in $\{j|l_j == s\}$ **do**
    Let $\delta_j = \mathbf{z}_j - \bar{\mathbf{z}}_s$
  end for
end for
$R = 0$
for $i = 1$ to N **do**
  $R = R + W_i \delta_j \delta_j^t$
end for
$\hat{R} = DRD$
$\hat{U} = eigs(\hat{R}, corank)$ % eigs produces the eigenvectors of the largest magnitude eigenvalues
$U = D^{-1}\hat{U}$

Algorithm 2 Iterative WNAP subspace training algorithm for a metric, D_i, with variable component $\tilde{D}_i$

Input: Data set $\{\mathbf{z}_i\}$ with N training vectors of dimension N_e, weights $\{W_i\}$, with speaker labels $\{l_i\}$, and an initial U of the desired corank
Output: Nuisance subspace, U

for all s in unique speakers in $\{l_i\}$ **do**
  Find $\bar{\mathbf{z}}_s$
  for all j in $\{j|l_j == s\}$ **do**
    Let $\hat{\delta}_j = \sqrt{W_j}\,(\mathbf{z}_j - \bar{\mathbf{z}}_s)$
  end for
end for
for i=1 to max iterations **do**
  for $j = 1$ to N **do**
    $\mathbf{x}_j = (D_j U) \backslash (D_j \hat{\delta}_j)$
  end for
  for $j = 1$ to N_e **do**
    $R_j = 0$, $\mathbf{v}_j = 0$
    for $k = 1$ to N **do**
      $R_j = R_j + W_k \tilde{D}_{k,j}^2 \mathbf{x}_j \mathbf{x}_j^t$
      $\mathbf{v}_j = \mathbf{v}_j + \tilde{D}_{k,j}^2 \hat{\delta}_{k,j} \mathbf{x}_j$
    end for
  end for
  for $j = 1$ to N_e **do**
    $U_j = (R_j + \epsilon I) \backslash \mathbf{v}_j$
  end for
end for
$U = \begin{bmatrix} U_1 & U_2 & \cdots & U_{N_e} \end{bmatrix}^t$

8. Experiments

Experiments were performed on the NIST 2006 speaker recognition evaluation (SRE) data set. Enrollment/verification methodology and the evaluation criterion, equal error rate (EER) and minDCF, were based on the NIST SRE evaluation plan [17]. The main focus of our efforts was the one conver-

Table 1: A comparison of compensation methods on the NIST SRE 2006 one conversation telephone train and test condition; W_i is the number of speech frames in the utterance

Compensation Method	Training Method/Metric	WNAP Projection	EER All (%)	minDCF All ($\times 100$)	EER Eng (%)	minDCF Eng ($\times 100$)
NAP	Eig, D	$Q_{U,D}$	3.87	2.05	2.49	1.34
NAP	Eig, D	Q_{U,D_i}	3.78	2.04	2.38	1.32
WNAP	Iter, D	$Q_{U,D}$	3.05	1.67	1.84	1.05
WNAP	Eig, D	$Q_{U,D}$	3.12	1.65	1.81	1.01
WNAP	Iter, D	Q_{U,D_i}	2.96	1.63	1.78	1.00
WNAP	Eig, D	Q_{U,D_i}	3.01	1.62	1.78	0.98
WNAP	Iter, D_i	$Q_{U,D}$	3.09	1.66	1.95	1.00
WNAP	Iter, D_i	Q_{U,D_i}	2.96	1.60	1.78	0.97

sation enroll, one conversation verification task for telephone recorded speech. T-Norm models and Z-Norm [18] speech utterances were drawn from the NIST 2004 SRE corpus. Results were obtained for both the English only task (Eng) and for all trials (All) which includes speakers that enroll/verify in different languages.

Feature extraction was performed using HTK [19] with 20 MFCC coefficients, deltas, and acceleration coefficients for a total of 60 features. A GMM UBM with 512 mixture components was trained using data from NIST SRE 2004 and from Switchboard corpora. The dimension of the nuisance subspace, U, was fixed at 128. A relevance factor of 0.01 was used for MAP adaptation.

For our experiments, we used weighting based upon our confidence in the parameter vector expansion—the number of speech frames in the utterance. The IPDF comparison function used was C_{GM} (3). Iterative methods were initialized with an equal weight NAP eigenvector solution, and 10 iterations were performed. Results are shown in Table 1. In the table, we use the following notation,

$$D = \left(\boldsymbol{\lambda}^{1/2} \otimes I_n\right) \Sigma^{-1/2},\ D_i = \left(\boldsymbol{\lambda}_i^{1/2} \otimes I_n\right) \Sigma^{-1/2} \quad (21)$$

where $\boldsymbol{\lambda}$ are the UBM mixture weights, $\boldsymbol{\lambda}_1$ are the mixture weights estimated from the enrollment utterance, and $\boldsymbol{\lambda}_2$ are the mixture weights estimated from the verification utterance. For the "nuisance free" estimate per speaker, $\bar{\mathbf{z}}_s$, we used the relevance MAP adapted vector obtained by combining sufficient statistics across all speaker sessions. An alternate strategy, used in [3], of taking the mean of the per session relevance MAP adapted mean vectors was not as accurate. Finally, we mention that we used the subspace of Algorithm 1 as a starting point for Algorithm 2.

An analysis of the results in Table 1 shows several trends. First, there is a substantial improvement in performance for C_{GM}, greater than 20% reduction in error rate, when going from NAP to WNAP. Second, the use of a variable metric, D_i, versus a fixed metric, D, appears to only provide minor (non-statistically significant) improvements in performance. Third, the eigenvector and iterative methods are essentially equivalent for a fixed metric, D. This property is extremely useful since we can leverage prior work [3] that uses iterative eigenvector methods such as Lanczos and KPCA to solve the WNAP optimization problem. Eigenvector methods in our experiments were about an order of magnitude faster than iterative methods. Fourth, we mention that the WNAP/IPDF combination has performance comparable to other systems such as JFA with linear scoring. In a system with a similar experimental setup to ours, see [8, 10] for more details, JFA has an EER/minDCF of 1.73/0.95 for the English condition.

9. Conclusions and Future Work

We have described a new method, WNAP, for reducing intraspeaker variability. WNAP incorporates several features including per utterance metrics and weighting of utterances. We demonstrated a fast eigenvector method for training the WNAP nuisance subspace. Significant performance improvements on a NIST SRE 2006 speaker recognition task were demonstrated. Future work includes exploring other weighting functions and application to other comparison functions and kernels (GLDS, high-level speaker recognition).

10. References

[1] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of ICASSP*, 2002, pp. 161–164.

[2] Alex Solomonoff, Carl Quillen, and William M. Campbell, "Channel compensation for SVM speaker recognition," in *Proceedings of Odyssey-04, The Speaker and Language Recognition Workshop*, 2004, pp. 57–62.

[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," in *Proceedings of ICASSP*, 2006, pp. I–97–I–100.

[4] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in *Proc. Odyssey04*, 2004, pp. 219–226.

[5] Andrew O. Hatch, Sachin Kajarekar, and Andreas Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. Interspeech*, 2006, pp. 1471–1474.

[6] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, 2008.

[7] Brendan Baker, Robbie Vogt, Mitchell McLaren, and Sridha Sridharan, "Scatter difference NAP for SVM speaker recognition," in *Lecture Notes in Computer Science*, vol. 5558, pp. 464–473. Springer, 2009.

[8] W. M. Campbell, Z. N. Karam, and D. E. Sturim, "Inner product discriminant functions," in *Advances in Neural Information Processing Systems 22*, Cambridge, MA, 2009, MIT Press.

[9] V. Wan and S. Renals, "SVMSVM: support vector machine speaker verification methodology," in *Proceedings of ICASSP*, 2003, pp. 221–224.

[10] Ondrej Glembek, Lukas Burget, Najim Dehak, Niko Brummer, and Patrick Kenny, "Comparison of scoring methods used in speaker recognition with joint factor analysis," in *Proceedings of ICASSP*, 2009.

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

[12] Robbie Vogt and Sridha Sridharan, "Explicit modelling of session variability for speaker verification," *Computer Speech and Language*, , no. 22, pp. 17–38, 2008.

[13] M. J. F. Gales, "Cluster adaptive training of hidden markov models," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 4, pp. 417–428, 2000.

[14] James C. Bezdek and Richard J. Hathaway, "Some notes on alternating optimization," in *Lecture Notes in Computer Science*, vol. 2275, pp. 187–195. Springer, 2002.

[15] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, John Hopkins, 1989.

[16] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller, "Kernel principal component analysis," in *Advances in Kernel Methods*, Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, Eds., pp. 327–352. MIT Press, Cambridge, Massachusetts, 1999.

[17] M. A. Przybocki, A. F. Martin, and A. N. Le, "NIST speaker recognition evaluations utilizing the Mixer corpora—2004,2005,2006," *IEEE Trans. on Speech, Audio, Lang.*, vol. 15, no. 7, pp. 1951–1959, 2007.

[18] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, pp. 42–54, 2000.

[19] J. Odell, D. Ollason, P. Woodland, S. Young, and J. Jansen, *The HTK Book for HTK V2.0*, Cambridge University Press, Cambridge, UK, 1995.