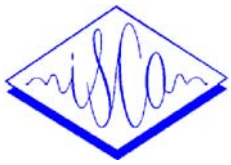

ODYSSEY04 - The Speaker and Language Recognition Workshop
Toledo, Spain
May 31 - June 3, 2004

Channel Compensation for SVM Speaker Recognition*

Alex Solomonoff, Carl Quillen, and William M. Campbell

MIT Lincoln Laboratory
Lexington, Massachusetts, USA
{als,cbq,wcampbell}@ll.mit.edu

Abstract

One of the major remaining challenges to improving accuracy in state-of-the-art speaker recognition algorithms is reducing the impact of channel and handset variations on system performance. For Gaussian Mixture Model based speaker recognition systems, a variety of channel-adaptation techniques are known and available for adapting models between different channel conditions, but for the much more recent Support Vector Machine (SVM) based approaches to this problem, much less is known about the best way to handle this issue. In this paper we explore techniques that are specific to the SVM framework in order to derive fully non-linear channel compensations. The result is a system that is less sensitive to specific kinds of labeled channel variations observed in training.

1. Introduction

Support Vector Machines (SVMs) have recently proved capable of providing good performance when applied in the speaker identification and verification domains [1, 2]. Not only is performance close to that of the best Gaussian Mixture Model (GMM) based systems, but these system possess substantial advantages in terms of computational cost, both in training and testing. When scores of SVM- and GMM-based systems are fused, the result is typically much better than either system alone, indicating in some sense that they provide information that is complementary.

These desirable properties motivate the study of ways to improve SVMs on their own account, and in the context of speaker recognition; an important place to start is the channel variation problem. By this we mean the performance degradations caused by variations in handsets and channel types that occur between testing and training in speaker recognition systems. For example, training data for an individual may be observed on one channel type (e.g. a carbon-button microphone telephone), and test data on another (e.g. cellphone). In this case impostors using carbon-button handsets are more likely to match the target speaker than usual, and the target speaker on a cell telephone is more likely to be incorrectly rejected.

In the case of GMM based systems, a variety of maximum likelihood model and feature space adaptations are known and have been studied in detail in the speech recognition and speaker-recognition literature (e.g. [3], [4]). Because we often combine an SVM-based speaker recognition system with a GMM system running in parallel, it's quite natural to consider GMM-derived feature transformations as a front-end to an SVM system. While this approach may have its merits, we constrain ourselves here to consider approaches inherently dependent on the SVM approach and the underlying mathematical machinery available in that setting.

In accordance with standard SVM practice, we make modifications of input feature vector data by mapping to a high dimensional space and then performing compensation. The outline of the paper is as follows. In Sections 2 and 3, we review support vector machines and their application to speech problems. Next, in Section 4, we consider unsupervised analysis of feature vector data using Kernel Principal Component Analysis (KPCA) [5]. KPCA enables us to visualize the basic structure of the data to suggest possible compensation methods. Section 5 covers the core approach of our paper which involves using projections to eliminate irrelevant directions in the expanded high-dimensional space. Section 6 illustrates our approach on the NIST extended data task.

2. Support Vector Machines

At the most basic level, SVMs are two-class hyperplane-based classifiers operating in a (usually) high-dimensional space related nonlinearly to the original (usually lower-dimensional) input space. Given an observation $x \in X$ and a kernel function K, an SVM, $f(x)$ is given by

$$\begin{aligned} f(x) &= \sum_{i=1}^{N} \lambda_i \xi_i K(x, x_i) + b \\ &= \sum_{i=1}^{N} \lambda_i \xi_i \phi(x) \cdot \phi(x_i) + b \end{aligned} \tag{1}$$

Note that we have assumed the Mercer condition [6]; that is, $K(x, y)$ is an inner product expressible as $\phi(x) \cdot \phi(y)$ where $\phi : x \mapsto y \in Y$ for some expansion space Y. We compare the output of the SVM in (1) to a threshold in order to produce a decision. The x_i, ξ_i, and $\lambda_i > 0$ are obtained through a training process. The x_i are called support vectors and the ξ_i are the target class values: $+1$ for in-class and -1 for out-of-class.

For the purposes of this paper, we will be studying various potential modifications to the kernel function $K(x, y)$ which increases its ability to be invariant to channel effects. We can view this as modifying the metric used by the SVM in order to destroy the ability of the SVM based system to tell apart different channel conditions. The hypothesis that we will make is that channel variations tend to lie in low-dimensional subspaces of Y, and that if we can project out of Y these dimensions, most of the speaker-dependent information in Y will be unaffected.

*This work was sponsored by the United States Air Force under Air Force contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

3. Speaker Recognition with Support Vector Machines

Speaker recognition fundamentally requires a decision based on an *utterance* x_i, which consists of a set of N_i feature vectors. As a result, methods for comparing sequences of vectors via a kernel will be important in our application. Several methods for comparing sequences of vectors have appeared in the literature. The Fisher kernel [7, 8] is a general method of comparing sequences based upon a generative model. Methods for comparing strings using automata theory are realized in rational kernels [9]. Finally, methods derived from the train/test process in speaker recognition are given in [2, 10]. We focus on the final set of methods since they are implemented specifically for speaker recognition, are computationally and parameter efficient, and are known to have low error rates.

After selection of an appropriate sequence kernel, speaker recognition is straightforward. Training is accomplished through a one vs. all strategy. For each enrolled speaker, we train with the target speaker's utterances labeled as $+1$ and utterances from a set of "background" speakers with a label of -1 using a standard SVM algorithm and the implemented sequence kernels. Testing is also straightforward. Given an input utterance, we evaluate the SVM using a sequence kernel and (1). The output $f(x)$ is compared to a threshold and the speaker is accepted or rejected based upon whether the value is above or below the threshold, respectively.

4. Visualization using KPCA

While SVM classifiers are nonlinear when their decision boundaries are viewed in the low-dimensional observation space X, they are linear in the related high-dimensional space Y. This allows us to use techniques like Principal Components Analysis (PCA), or Multi-Dimensional Scaling (MDS) from another point of view, to visualize what is occurring within these spaces.

PCA can be used to derive a low-dimensional projection of data. For example, for a d-dimensional projection, this amounts to taking the projection of the data on to the d principal eigenvectors of the covariance matrix, and results in the d-dimensional projection with minimum error in the L_2 norm. Given column vector data $\{p_i\}$, arranged in a matrix $A = [p_1, p_2, \cdots p_n]$, the covariance matrix C may be written

$$C = \frac{1}{n} A J J^t A^t = \frac{1}{n} A J A^t \tag{2}$$

where $J = J^2$ is the centering matrix that subtracts out the mean. This can be written in terms of $\mathbf{1}$, the column vector of all ones, as

$$J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathbf{t}}. \tag{3}$$

On occasion, it is easier to carry out PCA using a related method that does not use C, but the related matrix K where

$$K = J^t A^t A J \tag{4}$$

This has the same nonzero eigenvalues as equation (2) and related eigenvectors. Equation (4) is much more practical when the dimensionality of the data is much higher than the number of points n. Also, the matrix $A^t A$ consists only of inner products $p_i \cdot p_j$, and occasionally only inner-products of the data are available. Equation (4) is what is solved when using MDS, and it exactly matches our requirements in the case of SVM. In this case, the process is typically described as Kernel PCA (KPCA).

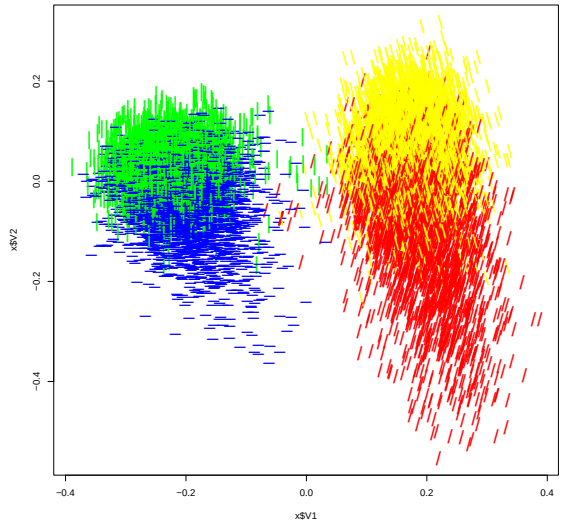

Figure 1: Two-dimensional representation of utterances from the 2003 NIST extended-data speaker evaluation using a phonetic-SVM kernel and KPCA. Green | = male-electret, Blue − = male-carbon-button, Yellow \ = female-electret, Red / = female-carbon-button.

Because we are great believers in visualizing data, and KPCA provides a simple method for viewing aspects of the high-dimensional space where the SVM classifier operates, we used this technique for several SVM speaker recognition systems. KPCA is performed by first calculating the kernel matrix $K(x_i, x_j)$ for all utterances x_i and x_j. The first few eigenvectors, $\alpha^1, \cdots, \alpha^k$ (corresponding to the largest eigenvalues) are then calculated using an iterative Lanczos method; we use ARPACK++ for this purpose [11, 12]. A k-dimensional representation of the ith utterance, r_i, is then obtained as

$$r_i = \begin{bmatrix} \alpha_i^1 & \alpha_i^2 & \cdots & \alpha_i^k \end{bmatrix}^t. \tag{5}$$

Figure 1 depicts the principal two eigenvectors of a phonetic SVM speaker recognition system described in [2]. Centering of the kernel was used as mentioned earlier and as described in detail in [5]. One should keep in mind that this picture is for an SVM system that doesn't directly use spectral features—it uses statistics derived from the output of speaker-independent phone recognizers in order to classify speakers. Despite the fact that one might expect these features to be somewhat decoupled from gender and channel effects, these two factors appear to describe the two principal eigenvectors of the covariance matrix.

This picture immediately tells us that our phonetically based system is far from independent of channel effects. But is also suggests the expedient of projecting out the confounding channel dependent dimensions, and the results of our attempts to do this are reported below in the remainder of this paper.

Figure 2 depicts eigenvectors 1 and 4 of a LP-cepstra-based SVM speaker recognition system. It also has significant separation between the two channels. See section 6 for a description of the corpora used by this system.

5. Channel Compensation using Projections

In this paper, we try to develop a modified kernel matrix for a SVM which projects out the effects of channel, or some other

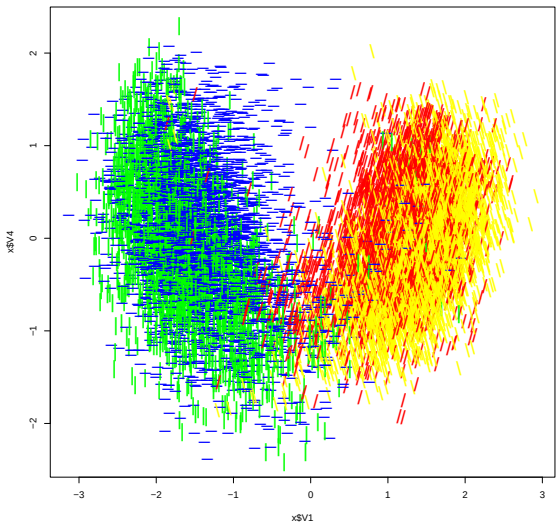

Figure 2: Two-dimensional representation of utterances from the 2003 NIST extended-data speaker evaluation using a cepstra-SVM kernel and KPCA. Green | = male-electret, Blue – = male-carbon-button, Yellow \ = female-electret, Red / = female-carbon-button.

confusing attribute. So, we seek an appropriate projection matrix P in the expansion space Y to do this job. We assume a priori that P's null space has a single dimension, spanned by the vector w, so P has the form

$$P = I - ww^t, \text{ with } \|w\| = 1,$$

and effectively we are trying to find a new expansion space mapping $\hat{\phi}(x) = P\phi(x)$. Our goal is then to find w.

We define the matrix K as the SVM kernel evaluated on the training points: $K_{ij} = \phi(x_i) \cdot \phi(x_j)$. If we define the expansion space data matrix A as

$$A = [\phi(x_0), \phi(x_1), \ldots, \phi(x_n)],$$

then $K = A^t A$ and our modified kernel matrix K' is defined by

$$K' = (PA)^t(PA) = K - Kv(Kv)^t,$$

where v is some vector satisfying $w = Av$. Since $\|w\| = 1$, this is the same as requiring that $v^t K v = 1$.

One criterion for constructing w (or P) is to minimize (over all possible P) the average distance, in the expansion space between a carbon button training point and an electret training point:

$$P = \operatorname*{argmin}_P \sum_{i \in elec, j \in cb} \|P(\phi(x_i) - \phi(x_j))\|^2.$$

A somewhat lengthy calculation described in section 5.4 shows that the w satisfying this criterion is obtained from the eigenvector having the largest eigenvalue of the generalized eigenvalue problem

$$KZKv = \lambda Kv. \tag{6}$$

with

$$Z = \text{diag}(W\mathbf{1}) - W,$$

and W is a weight matrix with positive elements for training point pairs we want to move together, and zero elements for training point pairs we don't want to move together. In our channel compensation case this gives us

$$W_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ have different channels} \\ 0 & \text{otherwise} \end{cases}$$

This problem is somewhat awkward since K is generally singular: if our training data is centered as is done for KPCA, then $A\mathbf{1} = 0$ and $K\mathbf{1} = 0$. Most algorithms for solving the generalized eigenproblem involve a Cholesky decomposition of the RHS matrix, processing the left side matrix with the Cholesky factor, and then solving an ordinary eigenvalue problem. This seems impossible or at least awkward in this case. A more convenient approach is to multiply both sides on the left by K^{-1} (the singularity of K does not cause any problem here) giving the nonsymmetric eigenvalue problem

$$ZKv = \lambda v. \tag{7}$$

The vector v needs to be normalized so that $v^t K v = 1$. This should happen automatically if the symmetric generalized eigenproblem (6) is solved, but not in the nonsymmetric case of equation (7).

5.1. Possible Modifications

5.1.1. Projecting Away Multiple Dimensions

We might want to use a set of m vectors to project out instead of just one, letting

$$P = I - \sum_i w_i w_i^t$$

be a projection matrix of rank $n - m$, and minimize the distance between channels over all such matrices. This corresponds to finding the m eigenvectors with largest eigenvalues of the same generalized eigenvalue problem (6).

5.1.2. Other Choices of Weightings

One possible drawback to the method of channel compensation presented here is that we try to minimize the average distance between *all* cross-channel pairs, not just the ones that would be very close except for channel differences. For example it tries to bring together pairs of utterances where the speakers sound very different, and happen also to be on different channels.

To eliminate this, the pairs being minimized might be weighted in different ways. One could minimize only the distance between point pairs that were both different channel *and* same speaker. A possible difficulty is that this might allow only a small number of pairs, requiring some kind of backoff or smoothing.

Another possibility is to cluster training speakers somehow and then only include pairs that were different channel and same cluster. Or we could assume that pairs that had small distances before channel compensation were probably similar-sounding speakers and minimize pairs that are different channel and small distance.

5.1.3. A Mixed Speaker-Channel Formulation

A related issue is the fact that the equation (6) tries to minimize cross-channel distances, but does nothing to increase cross-speaker distances, which might also result in an increase in performance. Presumably the eigenvalue problem that addresses

both of these issues is

$$K(\alpha Z_{\text{channel}} - \beta Z_{\text{speaker}})Kv = \lambda Kv, \tag{8}$$

where α and β are two positive weights and Z_{channel} is the channel difference matrix described earlier and Z_{speaker} is an analogous speaker difference matrix, based on the weight matrix

$$\left(W_{\text{speaker}}\right)_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ have different speakers} \\ 0 & \text{otherwise} \end{cases}$$

This eigenvalue problem is no more difficult to solve than (6).

5.2. Channel Compensation On Data Different From the Training Set

We usually want to do channel compensation on a set of points different from the ones we trained the projection vectors on. This requires a slightly different approach than what we have been describing.

Suppose $\{y_1, \ldots, y_m\}$ is a set of test points,

$$B = [\phi(y_1), \ldots, \phi(y_m)],$$

and $w = Av$ is the vector that is projected out in the expansion space. If

$$\begin{bmatrix} A^t \\ B^t \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix} = \begin{bmatrix} K & L \\ L^t & M \end{bmatrix}$$

Then the channel-compensated matrices are defined as follows:

$$K' = K - pp^t, \text{ where } p = Kv,$$

and

$$M' = M - qq^t, \text{ where } q = Lv,$$

and

$$L' = L - pq^t.$$

Here $M = B^tB$ is the kernel matrix for the test points and M' its channel-compensated version. The matrices $L = A^tB$ and L' are cross-corpus kernel matrices. L' doesn't have any application that we know of, but we describe it anyway.

In expansion space the channel-compensated test matrix is

$$B' = PB = (I - ww^t)B = B - wq^t.$$

5.3. Average Interpoint Distances

How does channel compensation effect the average distance between points in expansion space? Since a dimension is being projected away, we would expect that that average distance would get slightly smaller. Here we show this is true.

Without any channel compensation, the average interpoint distance is

$$\bar{\delta}^2 = \frac{1}{n^2} \sum_{i,j} \|\phi(x_i) - \phi(x_j)\|^2.$$

A bit of algebra gives us

$$n^2\bar{\delta}^2 = 2n\|A\|_F^2 - 2\|A\mathbf{1}\|^2.$$

If the points are centered, then $A\mathbf{1} = 0$, so

$$n^2\bar{\delta}^2 = 2n\|A\|_F^2 = 2n\text{tr}(JKJ), \tag{9}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. After the channel compensation a few lines of algebra give

$$n^2\bar{\delta}'^2 = 2n\left(\|A\|_F^2 - \|p\|^2\right) = 2n\left(\text{tr}(JKJ) - \|p\|^2\right),$$

and this is never bigger than the average distance without compensation.

5.4. Derivation of the Channel Compensation Equation (6)

This derivation is similar in style to the derivation of the Laplacian version of Locally Linear Embedding (LLE) equation (see [13]), but somewhat more elaborate.

Let W be an $n \times n$ weight matrix. We have discussed the two channel compensation case but other choices are possible, including negative values of W_{ij} for pairs of points we want to spread apart, or non-binary values of W_{ij}. The only requirement on W is that it be symmetric.

To make the equations less messy, we write $\phi(x_i) = \phi_i$. Then the figure of merit we want to minimize is

$$\delta = \sum_{i,j} W_{ij} \|P(\phi_i - \phi_j)\|^2. \tag{10}$$

Substituting in $P = I - ww^t$, $\|w\| = 1$, unfolding the vector norm and doing a couple lines of algebra gives

$$\delta = \sum_{i,j} W_{ij} \left(\|\phi_i - \phi_j\|^2 - \left(w^t(\phi_i - \phi_j)\right)^2\right) \tag{11}$$

Since the first term does not depend on w we ignore it, giving

$$\delta' = -\sum_{i,j} W_{ij} \left(w^t(\phi_i - \phi_j)\right)^2.$$

Unfolding the square and doing some algebra gives

$$\begin{aligned} \delta' &= -\sum_{i,j} W_{ij} \left((w^t\phi_i)^2 + (w^t\phi_j)^2 - 2w^t\phi_i w^t\phi_j\right) \\ &= -2\sum_{i,j} W_{ij} \left((w^t\phi_i)^2 - w^t\phi_i w^t\phi_j\right) \\ &= -2\sum_i \left(\sum_j W_{ij}\right) w^t\phi_i\phi_i^t w + \sum_{ij} W_{ij} w^t\phi_i\phi_j^t w. \end{aligned}$$

Now we re-express all this in terms of $A = [\phi_1, \ldots, \phi_n]$ and $s = W\mathbf{1}$:

$$\begin{aligned} \delta' &= -2w^tA\,\text{diag}(s)A^tw + 2w^tAWA^tw \\ &= 2w^tA(W - \text{diag}(W\mathbf{1}))A^tw \end{aligned} \tag{12}$$

We want to minimize this, subject to the constraint that $\|w\| = 1$, which is equivalent to finding the smallest eigenvalue of the symmetric eigenvalue problem

$$A(W - \text{diag}(W\mathbf{1}))A^tw = \lambda w. \tag{13}$$

This eigenvector problem occurs in expansion space, which we usually want to avoid working in. To work in kernel space, we can say that $w = Av$, and then we want to minimize

$$\delta' = 2v^tA^tA(W - \text{diag}(W\mathbf{1}))A^tAv \tag{14}$$

subject to the constraint that $\|Av\| = v^tKv = 1$, which corresponds to the generalized eigenvalue problem

$$K(W - \text{diag}(W\mathbf{1}))Kv = \lambda Kv,$$

which is the channel compensation equation (6).

5.5. Examination of δ_0 and δ'

It is also of interest to look at the first term of (11). This is the value of the objective function before channel compensation. A comparison of the first and second terms gives an indication of how effectively the compensation is working:

$$\begin{aligned}\delta_0 &= \sum_{i,j} W_{ij}\|\phi_i - \phi_j\|^2 \\ &= \mathrm{tr}A\big(\mathrm{diag}(W\mathbf{1}) - W\big)A^t.\end{aligned}$$

How do we express this in terms of K, in kernel space? Recall that we can reorder the matrices of a product when computing their trace: $\mathrm{tr}(MN) = \mathrm{tr}(NM)$, and so

$$\begin{aligned}\delta_0 &= \mathrm{tr}\big(\mathrm{diag}(W\mathbf{1}) - W\big)A^tA \\ &= \mathrm{tr}\big(\mathrm{diag}(W\mathbf{1}) - W\big)K \\ &= \mathrm{tr}K\big(\mathrm{diag}(W\mathbf{1}) - W\big).\end{aligned}$$

Let $\{\lambda_i\}_i$ be the eigenvalues of the generalized eigenvalue problem (15), ordered from most positive to most negative. Then

$$\delta_0 = \sum_{i=1}^{n} \lambda_i,$$

and

$$\delta = \delta_0 - \delta' = \sum_{i=2}^{n} \lambda_i.$$

5.6. Positive and Negative Eigenvalues

We have seen that the figure of merit δ is composed of two terms, δ_0 which is the figure of merit without any channel compensation, and δ' which is the change in δ from channel compensation. If δ' is not negative, then channel compensation has not improved our figure of merit.

If we choose a w for channel compensation according to (6), then $-\delta' = \lambda$, the principal eigenvalue of the equation. We need that eigenvalue to be positive for channel compensation to be successful.

If all elements of W are nonnegative, then $Z = \mathrm{diag}(W\mathbf{1}) - W$ is a diagonally semi-dominant matrix and therefore positive semidefinite.

From there we can use the Sylvester Inertia theorem (see [14]) to say that if K is strictly positive definite, then all eigenvalues of (6) are nonnegative.

So if $W_{ij} \geq 0\ \forall i, j$ then we can always successfully do channel compensation, or at least successfully improve δ.

If W has some negative elements, then (6) will probably have some negative eigenvalues. If W has enough negative elements and they are large enough, then (6) can have *no* positive eigenvalues. We have observed this, when doing the joint channel-speaker compensation, with large enough values of β.

Experimentally in this case, we have observed a reduction in the improvement achieved by channel compensation.

6. Experiments

We performed experiments based upon the 2003 NIST extended data task evaluation (using the "v1" lists). See [15] for a detailed description of the corpus. We considered the case of 1 utterance enrollment and only male speakers to understand channel compensation in a restricted case. A text-independent generalized linear discriminant kernel [1] using monomials of up to degree 3 was used. Input features were 18 LP cepstral coefficients (derived from 12 LP coefficients) and deltas. The standard channel-compensating measures were applied to the cepstra – per-utterance mean subtraction and RASTA normalization. The dimension of the SVM expansion space was 9139.

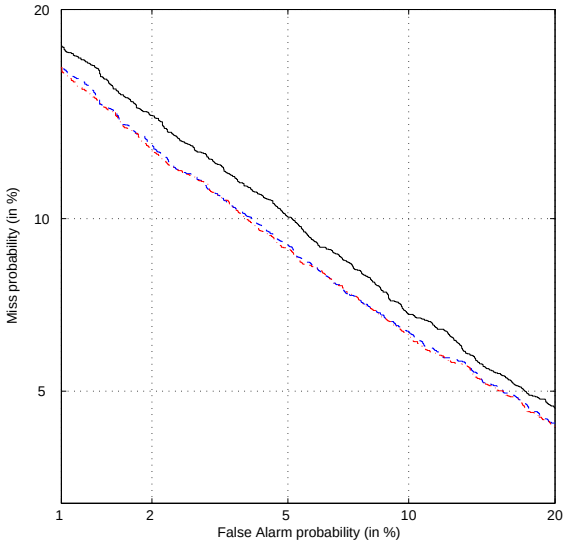

Figure 3: DET plot of the baseline (solid line) versus the new channel compensation approaches, rank 1 compensation (dashed) and rank 2 compensation (dash-dot).

Since the extended data task is landline telephone, we used carbon button and electret as our two channels. If the necessary label data are available, other choices of channel are possible. This would be an worthwhile topic of further work. For example, carbon button handsets differ greatly, and so calling each specific telephone number a different channel might be effective, at least for the carbon button handsets. It would also be interesting to see how the error rate differs between the same phone number and different phone number conditions on enrollment and test. We do this below, but just for carbon button and electret.

The training corpus is quite large, about 5600 utterances, each about 2.5 minutes long for a total of about 230 hours of speech. This entire corpus was used to train both the channel compensation projections and the SVM background model. It would be interesting to see how well the channel compensation works with a smaller training set, but in this SVM speaker ID context there is no reason not to use all of the data that is available for SVM training.

The test corpus consisted of 6190 true trials and 11544 false trials. The 95% confidence interval for all of the results is about $\pm 0.6\%$ absolute.

Figure 3 shows some initial experiments with the new channel compensation method where $\alpha = 1$ and $\beta = 1$ in (8). In the figure, the solid line shows a baseline approach using no compensation. The new approach is shown for both the rank 1 and rank 2 case. Note that the channel compensation provides a statistically significant decrease in the error rate. Also, note that rank 2 does not provide substantial improvement; this may be due to the fact that there are many superfluous dimensions which need to be removed.

Table 1 shows the results of different selections of α and β. A marginally better (but not statistically significant) improve-

Table 1: Comparison of EERs for different α and β

System	EER
Baseline	8.00%
$\alpha = 1, \beta = 0$, rank 1	7.51%
$\alpha = 1, \beta = 0$, rank 2	7.59%
$\alpha = 1, \beta = 1$, rank 1	7.43%
$\alpha = 1, \beta = 1$, rank 2	7.45%

Table 2: Comparison of EERs for different training and testing channels

System	Train Handset	Test Handset	EER
Baseline	CB	CB	9.79%
Rank 1			9.88%
Rank 2			9.84%
Baseline	ELEC	ELEC	3.40%
Rank 1			3.51%
Rank 2			3.56%
Baseline	CB	ELEC	8.18%
Rank 1			7.46%
Rank 2			7.35%
Baseline	ELEC	CB	9.54%
Rank 1			8.44%
Rank 2			8.44%

ment is obtained by using a weight on the speaker dependent metric in (8). Table 2 shows the results broken out by channel type for the case of $\alpha = 1$ and $\beta = 1$. The table clearly shows improvement in the cross-channel carbon-button (CB), electret (ELEC) cases. Minor degradation is seen for same channel conditions.

7. Conclusions

We have successfully demonstrated that SVM methods for visualization and channel compensations have substantial power in improving speaker verification. Kernel PCA was used to analyze sequence data and show clustering based upon gender and channel type. Further exploration of KPCA will no doubt lead to greater understanding of "speaker space." A novel channel compensation algorithm was then proposed and derived. Experiments on the NIST extended data task showed improvements in performance with the new method.

8. References

[1] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2002, pp. 161–164.

[2] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," in *Advances in Neural Information Processing 15*, 2003.

[3] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[4] A. Sankar and C. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 190–202, 1996.

[5] 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.

[6] Nello Cristianini and John Shawe-Taylor, *Support Vector Machines*, Cambridge University Press, Cambridge, 2000.

[7] Tommi S. Jaakkola and David Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. 1998, pp. 487–493, The MIT Press.

[8] V. Wan and S. Renals, "SVMSVM: Support vector machine speaker verification methodology," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2003, vol. 2, pp. 221–224.

[9] Corinna Cortes, Patrick Haffner, and Mehryar Mohri, "Rational kernels," in *Advances in Neural Information Processing Systems*, 2002, vol. 15.

[10] W. M. Campbell, "A SVM/HMM system for speaker recognition," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2003, vol. 2, pp. 209–212.

[11] F. A. M. Gomes and D. C. Sorensen, "ARPACK++: a C++ implementation of the ARPACK eigenvalue package," Tech. Rep. TR97729, Rice University, Houston, TX, USA, 1997.

[12] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, PA, 1998.

[13] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing 14*, T. G. Deitterich, S. Beck, and Z. Ghahramani, Eds., 2003.

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

[15] M. Przybocki and A. Martin, "The NIST year 2003 speaker recognition evaluation plan," http://www.nist.gov/speech/tests/spk/2003/index.htm, 2003.