NAP, WCCN, a New Linear Kernel, and Keyword Weighting for the HMM Supervector Speaker Recognition System

Howard Lei

TR-08-006

August, 2008

Abstract

We demonstrate the application of Nuisance Attribute Projection (NAP), Within-Class Covariance Normalization (WCCN), a new standard kernel, and keyword weighting for the keyword based HMM supervector speaker recognition system. On our development set (SRE04 8-side training), we achieve 22.6% and 16.2% EER improvements using NAP and WCCN respectively, a 19.5% EER improvement using NAP and WCCN jointly, and a 8.6% DCF improvement using keyword weighting. We also demonstrate the lack of effectiveness of gender separation in our NAP training, and counter-intuitive results using various modifications to the NAP technique. On our non-development set (SRE05 8-side training), we achieve a 5.3% EER and 18.6% DCF improvement using NAP and keyword-weighting, and a 13.7% EER and 12.3% DCF improvement using WCCN.

NAP, WCCN, a New Linear Kernel, and Keyword Weighting for the HMM Supervector Speaker Recognition System

Howard Lei

The International Computer Science Institute, Berkeley, CA, USA
The University of California, Berkeley, CA, USA
hlei@icsi.berkeley.edu

Abstract

We demonstrate the application of Nuisance Attribute Projection (NAP), Within-Class Covariance Normalization (WCCN), a new standard kernel, and keyword weighting for the keyword-based HMM supervector speaker recognition system. On our development set (SRE04 8-side training), we achieve 22.6% and 16.2% EER improvements using NAP and WCCN respectively, a 19.5% EER improvement using NAP and WCCN jointly, and a 8.6% DCF improvement using keyword weighting. We also demonstrate the lack of effectiveness of gender separation in our NAP training, and counter-intuitive results using various modifications to the NAP technique. On our non-development set (SRE05 8-side training), we achieve a 5.3% EER and 18.6% DCF improvement using NAP and keyword-weighting, and a 13.7% EER and 12.3% DCF improvement using WCCN.

Index Terms: speaker recognition, HMM supervectors, nuisance attribute projection, within-class covariance normalization, keyword weighting

1. Introduction

Speaker recognition has historically relied on low-level acoustic features with GMMs using a bag-of-frames approach for speaker discrimination and log-likelihood scoring [1]. Recently, the standard speaker recognition paradigm has shifted to GMM supervector-based methods, in which the means of Gaussian mixture models are collected into a feature vector, which are scored via SVMs [2]. This new paradigm has produced more effective speaker recognition systems, because speaker recognition is a binary classification task for which SVMs perform well. Moreover, the supervector-based approach allows for the application of various feature vector-based processing techniques and kernels that result in more effective SVM classifiers [3][4].

The use of keywords for text-constraining has also led to improvements in speaker recognition, as shown in [5]. In text-constraining, only portions of signals where certain texts (usually unigrams and bigrams) are spoken are used to construct the entire speaker recognition system. The advantages of text-constraining are to focus modeling power on more informative regions of speech, and to reduce negative effects that lexical variability may have on speaker recognition.

The HMM supervector speaker recognition system is based off of the keyword HMM system, which trains HMM models for each keyword (which we now refer to as keyword HMMs) to capture time-dependent information among the acoustic feature frames, and uses likelihood ratio scoring [6]. In the keyword HMM system, an HMM is trained using acoustic feature sequences within the boundaries of a set of keywords (keyword constraining) consisting of unigrams and bigrams. The HMM supervector system is an extension of this system, in that it takes the Gaussian mixtures means of all states of each keyword HMM to form the feature vector for the SVM classifier.

In this paper, we demonstrate the effectiveness of nuisance attribute projection (NAP) [3] and within-class covariance normalization (WCCN) [4] for our keyword-based HMM supervector speaker recognition system [7], along with the techniques of keyword weighting, gender separation as applied to NAP training, variations of NAP, and an intuitively appealing HMM supervector kernel. Note that [8] explicitly compares NAP and WCCN for an MLLR-SVM speaker recognition system.

This paper is organized as follows: Section 2 describes the database and preprocessing. Section 3 reviews the HMM supervector system. Section 4 reviews NAP and WCCN, explains other NAP techniques we've attempted, and provides a simple derivation of the WCCN projection. Section 5 describes the new HMM supervector kernel. Section 6 describes keyword weighting, and section 7 describes experiments and results. Section 8 provides a summary and conclusion of our findings.

2. Data and preprocessing

We used the Switchboard II and Fisher corpora for background model training, SRE04 for development, SRE05 for testing. Additionally, we used a different subset of the Switchboard II corpus to train NAP and WCCN projections. SRE04-05 are subsets of the MIXER conversational speech corpus, where two unfamiliar speakers speak for roughly 5 minutes. A conversation side (roughly 2.5 minutes for non-Fisher and 5 minutes for Fisher) contains speech from one speaker only. 8,804 conversation sides are used for SRE05, 2,843 from SRE04, 4,288 from Switchboard II, and 1,128 from Fisher. 1,553 Fisher and Switchboard II conversation sides are used as background conversation sides, where each speaker is represented by no more than one conversation side. Note that only the English telephone conversation sides are used. There are 7,336 total trials for SRE04 with 686 true speaker trials, and 20,683 trials for SRE05 with 2,072 true speaker trials.

We are provided with ASR decodings for all conversation sides by SRI, obtained via the DECIPHER recognizer [9]. We used MFCC features (C0-C19 plus deltas) with cepstral mean subtraction, obtained via HTK [10]. Note that we've attempted feature warping [11], but discovered no significant improvements after the application of NAP. Hence, feature warping is omitted for the results of this paper.

3. HMM supervector system

In this section, we discuss the basics of the HMM supervector system, which we modified from our original system in [7]. HMM training is done in the same manner as the traditional keyword HMM system [6], using a set of keywords. For each keyword, one background HMM is trained using MFCC features from the background conversation sides. For a given keyword, MFCC feature frames corresponding to all instances of the keyword are used to train the background keyword HMM. The observation distribution at each HMM state consists of a mixture of eight Gaussian components with diagonal covariances, and the number of keyword HMM states is a function of the number of phones and median number of MFCC frames comprising the keyword [6].

An HMM is then trained for all conversation sides of interest (i.e. target speaker, test, background, development) for each keyword. Keyword HMM training is done via MAP adaptation of the Gaussian mixture means from the corresponding background keyword HMMs using HTK, and if a keyword does not exist in a given conversation side, its keyword HMM parameters will be the same as those of the corresponding background keyword HMM. Note that only the Gaussian mixture means are adapted. Using the eight conversation side target speaker training condition greatly increases the probability that a keyword exists among the conversation sides used for training.

Next, we use the MAP adapted Gaussian mixture means of each keyword HMM for each conversation side as features in an SVM classifier. The Gaussian mixture mean vectors of each component of each state for each keyword HMM of a conversation side are concatenated to form a high-dimensional supervector, which represents the feature vector for the conversation side.

4. NAP and WCCN

4.1. NAP review and extensions

For a more detailed explanation of NAP, the reader should refer to [12] [3]. NAP is a technique for modifying the kernel distance between two features vectors via the removal of subspaces that cause undesired kernel variability [12]. Specifically, NAP for inter-session variability compensation requires the training of a feature vector projection matrix $P = I - VV^T$, where

$$V = \underset{V,||\text{all columns}(V)||_2=1}{\operatorname{argmin}} \sum_{i,j} W_{i,j}||P\Phi(x_i) - P\Phi(x_j)||_2^2 \quad (1)$$

where x_i is feature vector i, $\Phi(\cdot)$ is the mapping function from feature vector space to SVM space, and $W_{i,j} = 1$ if x_i and x_j belong to the same speaker, and $W_{i,j} = 0$ otherwise. Hence, the W matrix ensures that distances between features vectors of the same speaker are minimized, and the projection P projects away factors contributing to intra-speaker variability, such as channel or session variability.

We also attempted to maximize the inter-speaker variability, and counter-intuitively, to minimize inter-speaker variability, as was similarly done in [14] but within a slightly different framework. To minimize inter-speaker variability, set W_{ij} to 0 if conversation sides i and j belong to the same speaker and 1 otherwise. From now on, we denote this NAP approach as c-NAP. To maximize inter-speaker variability, we use the projection matrix $P = I + VV^T$, set W_{ij} to 0 if conversation sides i and j belong to the same speaker, and 1 otherwise, and maximize instead of minimize the objective in (1). From now on, we denote this NAP approach as m-NAP.

4.2. WCCN review and a simple alternate derivation

For a more detailed explanation of WCCN, the reader should refer to [4]. WCCN is a linear feature projection which aims to minimize the risk of misclassification of SVM classifiers for speaker recognition and related tasks [4]. The projection matrix A is obtained such that $A^TA = W^{-1}$, where $W = \rho W_s + (1-\rho)I$, W_s is the average of within-speaker covariance matrices of a set of impostor speakers, and ρ is a smoothing term. Note that this produces the following SVM kernel:

$$k(x_i, x_j) = x_i^T A^T A x_j = x_i^T W^{-1} x_j \quad (2)$$

While [4] provides detailed proofs, intuition, and analysis for deriving the WCCN projection based off of misclassification risk minimization, an alternative intuition for deriving the WCCN projection based off of the soft-margin SVM formulation [13] is as follows. The standard soft-margin SVM:

$$\begin{array}{lll} \min\limits_{w,b} & \frac{1}{2}w^Tw + C\sum_{i=1}^N s_i & \\ \textit{such that} & y_i(w^Tx_i + b) \geq 1 - s_i & i = 1, \ldots, N \\ & s_i \geq 0 & i = 1, \ldots, N \end{array} \quad (3)$$

where N is the total number of SVM training examples, and y_i equals 1 and -1 for positive and negative training examples respectively, can be derived as follows:

Assume that we are uncertain of the location of each training example x_i, and represent it as $x_i = \xi_i + \sigma\gamma$, where γ is a random vector with covariance matrix equal to the identity matrix. Then, for positive training examples, we wish to train w and b based off of the worst-case uncertainty as follows:

$$\min_{\gamma,||\gamma||_2=1} y_i(w^Tx_i + b) \geq -s_i \Rightarrow \quad (4)$$

$$y_i(w^T\xi_i + \min_{\gamma,||\gamma||_2=1} \sigma w^T\gamma + b) \geq -s_i \Rightarrow \quad (5)$$

$$y_i(w^T\xi_i - \sigma||w||_2 + b) \geq -s_i \Rightarrow \quad (6)$$

$$y_i(w^T\xi_i + b) \geq \sigma||w||_2 - s_i \quad (7)$$

Note that an analogous derivation can be made for negative training examples. We wish to maximize the "margin" surrounding each training example by maximizing σ. If we set $\sigma||w||_2 = 1$, then our SVM problem which minimizes $\frac{1}{2}w^Tw$ also maximizes σ. After a change of variables from ξ_i to x_i, we obtain the soft-margin SVM.

Now, if we assume that the covariance matrix of the uncertainty of our training examples is Σ, we can express our training examples as $x_i = \xi_i + \sigma\Sigma^{\frac{1}{2}}\gamma$. By going through the same mathematical procedure as above, we obtain:

$$y_i(w^T\xi_i + b) \geq \sigma||\Sigma^{\frac{1}{2}}w||_2 - s_i \quad (8)$$

Setting $\sigma||\Sigma^{\frac{1}{2}}w||_2 = 1$, and minimizing $||\Sigma^{\frac{1}{2}}w||_2$, we obtain the following SVM problem:

$$\begin{array}{lll} \min\limits_{w,b} & \frac{1}{2}w^T\Sigma w + C\sum_{i=1}^N s_i & \\ \textit{such that} & y_i(w^Tx_i + b) \geq 1 - s_i & i = 1, \ldots, N \\ & s_i \geq 0 & i = 1, \ldots, N \end{array} \quad (9)$$

Taking the dual of this new SVM convex optimization problem, we obtain the following kernel: $k(x_i, x_j) = x_i^T\Sigma^{-1}x_j$. Note that this kernel is similar to the WCCN kernel:

$k(x_i, x_j) = x_i^T W^{-1} x_j$, except that W is the smoothed and averaged empirical within-speaker covariance matrix, whereas Σ is the non-empirical covariance matrix of a set of training examples. Hence, we can think of WCCN as training an SVM to better fit the covariances of the data.

In practice, we implement WCCN using the approach described in [15] to avoid dealing with covariance matrices of large feature vector dimensions (∼20,000 for our HMM supervector system). Specifically, we perform mean and variance normalization on our data, perform kernel PCA on the data keeping the top N dimensions, and train the WCCN projection on the N dimensions of the PCA-projected data. To apply the projection, we again follow the approach in [15]. First, we perform kernel PCA on our data. Then, we apply the WCCN projection on the PCA-projected data. Lastly, we concatenate the WCCN-projected data with the PCA-complement [16] of the original data to form our final feature vector.

5. HMM linear kernel

In this section, we outline our derivations for a different linear kernel for HMM supervectors than the one described in [17]. Unlike the linear kernel in [17], which accounts only for the Gaussian mixture weights, means, and covariances, our kernel accounts for all keyword HMM model parameters. Similar to the approach in [2], we start with an upper-bound for the KL-distance of two models M_1 and M_2, where each model contains the GMM means, variances, and weights, state transition probabilities, and initial state probability distributions of our keyword HMM models. An upper bound is given in [18], for models M and $\hat{M}$:

$$D(M||\hat{M}) \leq D(\pi||\hat{\pi}) + \pi^T (I - A)^{-1} (D(A||\hat{A}) + D(b||\hat{b})) \quad (10)$$

where π represents the initial state probability distribution for model M, A is the transition probability matrix for model M, $D(A||\hat{A})$ is a vector where each row is the KL-distance of the transition probabilities for a given state, and $D(b||\hat{b})$ is a vector where each row is the KL-distance of GMM distributions for a given state. When only the Gaussian mixture means differ, we have:

$$D(M||\hat{M}) \leq \pi^T (I - A)^{-1} D(b||\hat{b}) \quad (11)$$

We can upper bound each component of $D(b||\hat{b})$ as in [2]:

$$D(b||\hat{b})_j \leq \frac{1}{2} \sum_{i=1}^{K} w_{ij} (m_{ij} - \hat{m}_{ij})^T \Sigma_{ij}^{-1} (m_{ij} - \hat{m}_{ij}) \quad (12)$$

where K is the number of Gaussian mixtures per state, m_{ij} and w_{ij} are the Gaussian mixture mean vector and weight respectively for mixture i and state j.

Now, because we use only left-to-right HMMs, $\pi = [1, 0, \ldots, 0]$, where the probability that the HMM starts in the first state is 1. Hence, the term $\pi^T (I - A)^{-1}$ is simply the first row of the matrix $(I - A)^{-1}$. Because the HMMs have no skips, $A(i, j) = 0$ for $j \neq i$ or $j \neq i + 1$, the matrix A is only non-zero in its diagonal and first off-diagonal above the diagonal. Hence, $(I - A)$ is a tridiagonal matrix, and according to [19], which shows the explicit inverse of a tridiagonal matrix, we have, after some additional algebraic manipulations:

$$\pi^T (I - A)^{-1} = \left[\frac{1}{1 - A_{11}}, \ldots, \frac{1}{1 - A_{JJ}} \right] \quad (13)$$

Note that this is a non-negative vector. Combining (11), (12), and (13), we have, for an upperbound of $D(M||\hat{M})$:

$$\sum_{j=1}^{J} \sum_{i=1}^{K} \frac{1}{1 - A_{jj}} w_{ij} (m_{ij} - \hat{m}_{ij})^T \Sigma_{ij}^{-1} (m_{ij} - \hat{m}_{ij}) \quad (14)$$

Using the polarization identity as in [2], we obtain the kernel for HMM supervectors m and $\hat{m}$:

$$K(m, \hat{m}) = \sum_{j=1}^{J} \sum_{i=1}^{K} \frac{1}{1 - A_{jj}} w_{ij} m_{ij}^T \Sigma_{ij}^{-1} \hat{m}_{ij} \quad (15)$$

Note that this kernel is only for one keyword HMM model. This kernel is intuitively appealing, since each component of a supervector is weighted by its Gaussian mixture weight, along with a measure of the relative likelihood that the HMM stays in its state.

6. Keyword selection and weighting

We've reduced the number of keywords (38) used in [7], and found that we achieve better performance using 16 of the most frequent keywords. The 16 keywords are the following: *but, have, just, like, not, really, right, so, that, think, uh, uhhuh, um, was, yeah, you_know*. Note that many of these keywords (i.e. *like, uh, uhhuh, um, yeah*) are used as fillers and back-channels.

To emphasize keywords that perform well individually, we've weighted the supervector components belonging to each keyword by the average EER of the keywords divided by the individual EER of the keyword in our SRE04 development set. Note that this technique is applied immediately prior to SVM training. It can not be applied after WCCN, as the PCA and PCA-complement operations of WCCN disassociate feature vector components from their keyword labels.

7. Experiments and Results

We've used 2,743 Switchboard II conversation sides to train the NAP and WCCN projections, 4,296 Fisher and Switchboard II conversation sides to train the PCA projection for WCCN. In addition, prior to applying any of the projections, we've rank-normalized our HMM supervector features based off of their rank amongst 1,553 Fisher and Switchboard II conversation sides, which are distinct from the 2,743 Switchboard II conversation sides used to train the projections. Our best results for our development set uses a WCCN smoothing factor of 0.5, PCA projection dimension of 3,000, 600 nuisance dimensions for regular NAP, and 39 projection dimensions for m-NAP and c-NAP. We've also trained gender-dependent NAP projections using 1,855 male and 1,998 female Switchboard II conversation sides, and applied NAP on top of WCCN. From now on, we denote the gender-separated NAP as gs-NAP.

We've also compared the performance of our new HMM linear kernel to the simple dot-product kernel, which we refer to as the dot kernel. Table 1 shows our best results for NAP, WCCN, and keyword weighting (KW) on the SRE04 8-side English development corpus with 7,336 trials. Also shown are results on the SRE05 8-side English telephone test corpus with 20,683 trials, which are obtained using updated ASR decodings (provided by SRI) and system implementation. The baseline system refers to the HMM supervector system with rank-normalization. The first two rows of table 1 shows that the linear kernel gives no improvement over the dot kernel on

system	kernel	KW	SRE04 EER	SRE04 min DCF
baseline	dot	no	4.52%	0.0178
baseline	linear	no	4.52%	0.0178
baseline+NAP	dot	no	3.50%	0.0151
baseline+NAP	dot	yes	3.50%	0.0138
baseline+gs-NAP	dot	yes	3.64%	0.0144
baseline+c-NAP	dot	no	4.08%	0.0153
baseline+m-NAP	dot	no	5.69%	0.0252
baseline+WCCN	dot	no	3.79%	0.0158
baseline+WCCN +NAP	dot	no	3.64%	0.0150
system	kernel	KW	SRE05 EER	SRE05 min DCF
baseline	dot	no	3.86%	0.0122
baseline	dot	yes	3.81%	0.0124
baseline+NAP	dot	yes	3.61%	0.0101
baseline+WCCN	dot	no	3.33%	0.0107

Table 1: *8-side training results on SRE04 development set and SRE05.*

SRE04. This may be because HMM model components which contribute little to log-likelihood scoring (i.e. components from states with low input transition probabilities) are still significant in SVM classifiers. Hence, the dot kernel is used for the rest of our experiments. Table 1 also shows that NAP with no keyword-weighting gives a 22.6% EER decrease over the baseline (3.50% EER vs. 4.52% EER) and a 15.2% decrease in min DCF (0.0178 vs. 0.0151). Keyword-weighting does not decrease EER, but decreases DCF by 8.6% (0.0151 vs. 0.0138). WCCN is not as effective as NAP for both EER and DCF. It gives a 16.2% EER decrease over the baseline, and a 11.2% decrease in DCF. WCCN+NAP decreased EER by 19.5% and DCF by 15.7% over the baseline.

What's interesting are the effects of gs-NAP, c-NAP, and m-NAP. In each case, the result runs counter to intuition. gs-NAP resulted in a 4.0% increase in EER and 4.4% increase in DCF compared to regular NAP with keyword weighting. This may be due to the fact that inter-session variability is invariant with respect to gender, and that fewer NAP training conversation sides are used per gender. c-NAP and m-NAP each give the opposite of what's expected. c-NAP, for which we've attempted to minimize inter-speaker variability, decreased EER by 9.7% and DCF by 14.0% over the baseline. m-NAP, for which we've attempted to maximize inter-speaker variability, increased EER by 25.9% and DCF by 41.6% over the baseline. These results are similar to the results in [14]. More investigation will be needed to determine why c-NAP and m-NAP behave as they do.

NAP is less effective on SRE05. NAP decreases EER by 5.3% and DCF by 18.6% w/ keyword weighting, while WCCN decreases EER by 13.7% and DCF by 12.3%.

8. Conclusion

We've demonstrated in this paper that NAP and WCCN are both effective techniques for improving our HMM supervector system. We've also introduced a new standard HMM supervector linear kernel that utilizes all HMM parameters. However, we find that this new kernel does not give significant improvements over the dot product kernel. Lastly, we've demonstrated the counter-intuitive results for gender separated NAP, m-NAP, and c-NAP. Future work should investigate and explain the effects of these modified NAP techniques for our HMM supervector system.

9. Acknowledgements

The author wishes to thank Andreas Stolcke of SRI for providing speech recognition decodings. This research is funded by NSF grant number 0329258.

10. References

[1] Reynolds, D.A., Quatieri, T.F., Dunn, R., "Speaker Verification using Adapted Gaussian Mixture Models", in Dig. Signal Process., Vol. 10, no. 1-3, pp. 19–41, 2000.

[2] Campbell, W.D., Sturim, D.E., Reynolds, D.A., "Support Vector Machines using GMM Supervectors for Speaker Verification", in IEEE Signal Processing Letters, Vol. 13, pp. 308-311, 2006.

[3] Campbell, W.D., Sturim, D.E., Reynolds, D.A., Solomonoff, A., "SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP Variability Compensation", in Proc. of ICASSP, 2006.

[4] Hatch, A.O., "Generalized Linear Kernels for One-Versus-All Classification: Application to Speaker Recognition", in Procc of ICASSP, 2006.

[5] Sturim, D., Reynolds, D., Dunn, R., Quatieri, T., "Speaker Verification using Text-Constrained Gaussian Mixture Models", in Proc. of ICASSP, Vol. 1, pp. 677–680, 2002.

[6] Boakye, K., "Speaker Recognition in the Text-Independent Domain Using Keyword Hidden Markov Models", Masters Report, University of California at Berkeley, 2005.

[7] Lei, H., Mirghafori, N., "Word-Conditioned HMM Supervectors for Speaker Recognition", in Proc. of Interspeech, 2007.

[8] Kajarekar, S., Stolcke, A., "NAP and WCCN: Comparison of Approaches using MLLR-SVM Speaker Verification System", in Proc. of ICASSP, 2007.

[9] Kajarekar, S., Ferrer, L., Venkataraman, A., Sonmez, K., Shriberg, E., Stolcke, A., Gadde, R.R., "Speaker Recognition using Prosodic and Lexical features", in Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop, pp. 19-24, 2003.

[10] HMM Toolkit (HTK): http://htk.eng.cam.ac.uk

[11] Pelecanos, J., Sridharan, S., "Feature Warping for Robust Speaker Verification", in Proc. of Odyssey, pp. 213–218, 2001.

[12] Solomonoff, A., Campbell, W. M., Boardman, I., "Advances in Channel Compensation for SVM Speaker Recognition," in Proc. of ICASSP, 2005.

[13] Vapnik, V., "The Nature of Statistical Learning Theory", Springer, 1999.

[14] Vogt, R., Kajarekar, S., Sridharan, S., "Discriminant NAP for SVM Speaker Recognition", in Proc. of Odyssey, 2008.

[15] Hatch, A.O., Kajarekar, S., Stolcke, A., "Within-class Covariance Normalization for SVM-based Speaker Recognition", in Proc. of ICSLP-Interspeech 2006.

[16] Kajarekar, S., "Four Weightings and a Fusion: a Cepstral-SVM System for Speaker Recognition", in Proc. of ASRU, 2005.

[17] Dong, C., Dong, Y., Li, J., Wang, H., "Support Vector Machines Based Text Dependent Speaker Verification Using HMM Supervectors", in Proc. of Odyssey, 2008.

[18] Silva, J., Narayanan, S., "Upper Bound Kullback-Leibler Divergence for Hidden Markov Models with Application as Discrimination Measure for Speech Recognition", in Proc. of IEEE International Symposium on Information Theory, 2006.

[19] Schlegel, P., "The Explicit Inverse of a Tridiagonal Matrix", in Mathematics of Computation, Vol. 24, no. 111, 1970.