Phone-based Cepstral Polynomial SVM System for Speaker Recognition

Sachin S. Kajarekar

SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025

sachin@speech.sri.com

Abstract

We have been using a phone-based cepstral system with polynomial features in NIST evaluations for the past two years. This system uses three broad phone classes, three states per class, and third-order polynomial features obtained from MFCC features. In this paper, we present a complete analysis of the system. We start from a simpler system that does not use phones or states and show that the addition of phones gives a significant improvement. We show that adding state information does not provide improvement on its own but provides a significant improvement when used with phone classes. We complete the system by applying nuisance attribute projection (NAP) and score normalization. We show that splitting features after a joint NAP over all phone classes results in a significant improvement. Overall, we obtain about 25% performance improvement with polynomial features based on phones and states, and obtain a system with performance comparable to a state-of-the-art SVM system.

Index Terms: Speaker recognition, feature extraction, pattern recognition.

1. Introduction

The most commonly used statistical modeling approaches for audio-based speaker recognition are the Gaussian mixture model (GMM) [1] and support vector machine (SVM) [2]. The former uses features based on a window of speech. The latter uses features estimated from the complete speech file. Researchers have explored different ways of estimating features from the complete speech file, and some successful approaches are based on Gaussian mean supervectors [3] and average polynomial vectors [2]. In this paper, we use average polynomial vectors and estimate them based on phones (and states within phones) obtained from an automatic speech recognition (ASR) system.

We have used polynomial features with SVMs in our earlier work [4]. We used two types of polynomial features – mean and mean divided by standard deviation. Division by standard deviation serves as a preprocessing step so that the kernel is redefined as inner product normalized by those statistics. Overall, mean divided by standard deviation polynomial features performed the best and their scores were combined with those from mean polynomial features to produce the final score.

We experimented with conditioning the polynomial features based on the spoken phone. The phones were grouped into three broad categories – vowels and diphthongs (VD), glides and nasals (GN) and obstruents (OB). The phone boundaries were obtained from our ASR system. The choice of categories was based on our earlier work [5]. Results showed that the VD subsystem gave the best performance, followed by GN and OB. The linear combination of scores from these three systems resulted in a significant improvement over individual systems and over a system that used all the classes together.

We also experimented with features based on the states within each phone category. Since the ASR system uses three states, the number of features increased threefold. These features significantly improved performance. However, the resulting system became very complex.

In this paper, we report work in two areas. First, we deconstruct the system to understand where the improvement is really coming from. This leads to simplification of the overall system. Second, we apply a intersession variability (ISV) normalization technique, nuisance attribute projection (NAP) [6], and show improvement in performance.

2. Datasets

We report results on NIST 2005 and 2006 speaker recognition evaluation (SRE) sets. From these sets, we use the 1conv4w training condition and the 1conv4w testing condition, where 1conv4w refers to approximately 2.5 minutes of speech from one side of a 5 minute conversation from the Mixer corpus [7].

The NIST 2005 common condition has 2967 test waveforms and 584 models, resulting in 30427 trials. The NIST 2006 common condition (from NIST release version 4) has 2692 test waveforms and 517 models, resulting in 24013 trials. The purpose of using two datasets is to show generalization of results and to perform score-level combination. When combining systems (at the score-level), we use a simple weighting of the scores. The weights are trained on 2005 data and are applied on 2006 data.

Results are presented as equal error rate (EER) and minimum normalized decision cost function (DCF) as defined by NIST. Please refer to NIST evaluation plans for more detail [8].

3. Baseline systems

The baseline system is similar to what was described in [4]. This system was a score-level combination of four subsystems. These systems used SVM for speaker modeling and differed in the way features were obtained.

Figure 1 shows the basic feature extraction algorithm. Thirteen cepstral coefficients are estimated from a 25 ms window of speech. Cepstral mean subtraction is applied to them and they are appended with delta and double delta coefficients to get a 39 dimensional feature vector. The mean and standard deviation of this vector are computed (per element) over the speech utterance and each vector is z-normed with these statistics. This results in 39 dimensional normalized feature vector.

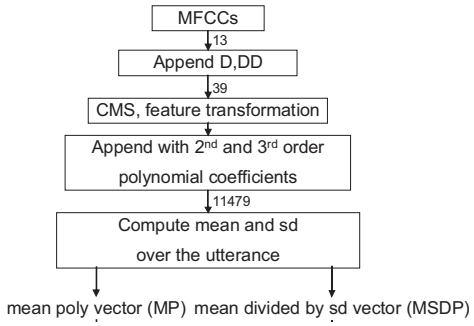

Figure 1 Feature extraction block diagram (with dimensionality shown on the arrows)

First-, second-, and third- order polynomial coefficients are appended to each feature vector. This results in an 11479 dimensional vector. The mean and standard deviation of this vector are computed over the speech utterance. The mean polynomial vector is referred to as MP. Each mean value is divided by its corresponding the standard deviation. This normalizes the linear kernel by the standard deviation of each feature. The resulting feature is referred to as the "mean divided by standard deviation polynomial" (MSDP) vector.

In the baseline system [4], MP was used in addition to the MSDP vector. Further each feature vector was projected onto the eigenvectors obtained from the background (or impostor) speakers and onto the eigenvectors obtained from the feature space that is orthogonal to the background speakers.

In this paper, we use only MSDP features because they performed better than MP features. Although the latter gave improvement in combination, we observed that this improvement was significantly reduced with phone classes. With three broad phone classes, the improvement was almost negligible.

Further, we use the features without any transformation, mainly because we want to measure the effect of NAP on these features. In our experiments, we observed that NAP and our previously proposed transformation technique interact with each other. We will explore these interactions in another paper.

Table 1 shows performance of this polynomial feature-based system. Note that the system does not use any information about phones. For the sake of consistency, we also refer to the polynomial feature-based system as "one state and one class" or "1s1c" system. Also note that NAP is not applied to the features here.

Table 1 Baseline with 1s1c – 1state and 1class – features

System	SRE05		SRE06	
	%EER	DCF	%EER	DCF
1state, 1class	8.12	0.300	6.74	0.295

4. Incorporating Phone-specific Information

We extend the baseline polynomial system using information from the ASR system. The ASR system models 40 phones with a 3-state hidden Markov model (HMM). If we estimate polynomial features for each phone then we will have data scarcity issues. We overcome this problem by clustering phones. Based on our previous work we group phones into three broad classes – vowels and diphthongs, glides and nasals, and obstruents. The choice of classes is based on similarity in the acoustic space.

In the following subsections, we extend the basic polynomial feature by first modeling states across all phones and then by using three broad classes with a single state. This will show the importance of states versus classes. Finally, we use both the classes and states and show overall performance.

4.1. Three states and one class (3s1c) features

Table 2 shows results after adding state-based features into the system. Comparing the performance with Table 1, it can be seen that the states do not add significantly different information, and increasing the features threefold does not improve performance. One reason could be that the states are specific to the phones and by averaging state information across phones we have lost that information.

Table 2 Results with 3s1c (3 states, 1 class) features

System	SRE05		SRE06	
	%EER	DCF	%EER	DCF
3 states, 1 class	8.36	0.314	6.53	0.288

4.2. One state and three classes (1s3c) features

Table 3 shows performance after adding features based on the three broad phone categories. The resulting features have the same cardinality as the state-based features. We investigate two types of modeling approaches. In the first case, the polynomial features based on phone classes are modeled independently. The resulting scores are combined with equal weight to generate the final output. In the second case, the entire feature vector is modeled jointly.

Results show that joint modeling does not improve the features. However, independent modeling of the classes followed by score-level combination (first row of Table 3) does improve performance in comparison with Table 1.

Table 3 Results with 1s3c (1 state, 3 classes) features

Modeling Classes	SRE05		SRE06	
	%EER	DCF	%EER	DCF
Independent	7.72	0.282	6.04	0.283
Joint	9.65	0.355	6.48	0.295

4.3. Three states and three classes (3s3c) features

Table 4 shows performance of polynomial features based on three classes and three states. Again, two sets of experiments are performed with independent and joint modeling of the classes. As with the results from Table 3, independent modeling of the phone classes gives significantly better performance than joint modeling, and the best performance overall. Comparing Table 1 and Table 4, we see that about 25% improvement is obtained on both 2005 and 2006 SREs by incorporating phone and state information in the polynomial features.

Table 4 Results with 3s3c (3 states and 3 classes) features

Modeling Classes	SRE05		SRE06	
	%EER	DCF	%EER	DCF
Independent	6.23	0.251	4.96	0.250

Joint	6.83	0.274	5.57	0.261

5. Intersession Variability Compensation

The most common approach to intersession variability compensation (ISV) with SVMs is nuisance attribute projection (NAP) [6]. The nuisance attributes are the directions in the feature space that model within-speaker variability. The variability is modeled as within-speaker covariance, and eigen decomposition of this matrix is performed. The leading N eigenvectors are referred to as nuisance attributes because they model variations across different sessions of the same speaker that come from differences in communication channel, handset, phonetic content, and so on.

We used NIST 2004 speaker recognition evaluation data for computing NAP, and N was determined empirically based on NIST 2005 evaluation data.

Table 5 Performance of different systems after NAP

System (N)	SRE05		SRE06	
	%EER	DCF	%EER	DCF
1state, 1class (128)	6.91	0.269	5.44	0.268
3state, 1class (32)	7.16	0.279	5.39	0.265
1state, 3class – Independent (64)	5.95	0.240	5.01	0.246
1state, 3class – Joint (256)	6.43	0.248	5.02	0.248
3state, 3class – Independent (128)	5.67	0.220	4.21	0.218
3state, 3class – Joint (128)	6.27	0.233	4.38	0.227

Table 5 shows the performance of different systems after NAP. Apart from the obvious improvement obtained after NAP, there is one interesting result. For three-class systems (1s3c and 3s3c), the performance difference between independent and joint modeling is much smaller after NAP. For example, performance of 1 state and 3 class features with independent and joint models differed by about 10% before NAP. After NAP, the difference is less than 4%. This suggests that NAP is taking advantage of joint modeling of phone classes by exploiting the correlations across classes.

5.1. Splitting phone-based features after joint NAP

To summarize the results with phone classes, we see that independent modeling of phone classes gives improvement over joint modeling without NAP. With NAP, independent modeling does not offer any advantage over joint modeling. We analyze this result in detail with NIST 2006 data.

Table 6 Detailed analysis of 1s3c system before and after NAP

Classes	Before NAP		After NAP (N)	
	%EER	DCF	%EER	DCF
1) VD	7.45	0.344	6.53	0.304
2) GN	8.79	0.392	8.95	0.371
3) OB	10.68	0.455	9.22	0.443
1)+2)+3) (Independent)	6.04	0.283	5.01 (64)	0.246
All classes (Joint)	6.48	0.295	5.02 (256)	0.248

Table 6 and Table 7 show two interesting trends with NAP. First, glides and nasals (GN) show the least improvement in DCF after NAP. In fact, EER for GN is slightly worse after NAP. For 1s3c features there is no improvement in EER for GN. It is not clear why NAP would favor other classes over them.

Table 7 Detailed analysis of 3s3c system before and after NAP

Classes	Before NAP		After NAP (N)	
	%EER	DCF	%EER	DCF
1) VD	6.20	0.300	5.45	0.269
2) GN	6.74	0.333	6.25	0.298
3) OB	9.66	0.423	8.41	0.408
1)+2)+3) (Independent)	4.96	0.250	4.21	0.218
All classes (Joint)	5.57	0.261	4.38	0.227

Second, NAP gives more improvement with joint modeling than with independent modeling. Since independent modeling is better before NAP, the performance of two models becomes very similar after NAP. One of the reasons is that with the independent model, the performance of all phone classes improves equally. This leads to a hypothesis that joint NAP takes advantage of the across-class correlations thus leading to greater improvement across all classes. We test this hypothesis by splitting the features after NAP.

Table 8 Splitting 1s3c features after joint NAP

Classes	After NAP			**Splitting after Joint NAP**	
	%EER	DCF		%EER	DCF
1) VD	6.53	0.304		6.15	0.284
2) GN	*8.95*	*0.371*		*7.98*	*0.337*
3) OB	9.22	0.443		8.95	0.386
1)+2)+3) (Independent)	5.01 (64)	0.246	↔	4.75	0.230
All classes (Joint)	5.02 (256)	0.248			

Table 9 Splitting 3s3c features after joint NAP

Classes	After NAP (128)			**Splitting after Joint NAP**	
	%EER	DCF		%EER	DCF
1) VD	5.45	0.269		5.06	0.258
2) GN	*6.25*	*0.298*		*5.99*	*0.282*
3) OB	8.41	0.408		7.60	0.367
1)+2)+3) (Independent)	4.21	0.218	↔	4.10	0.209
All classes (Joint)	4.38	0.227			

We model the features after NAP independently and report the results in Table 8 and Table 9. These results show that joint NAP does improve per-phone-class performance more than independent NAP. It also improves the performance of the GN class, which showed the smallest improvement from independent NAP (Table 6 and Table 7). The performance of the score-level combination is not improved as much as the performance of individual phone

classes, mainly because we used equal weights for the combination. The performance might be better if the weights were tuned further.

5.2. Score normalization

We complete the results by applying score normalization [9] after NAP. We look at three techniques – ZNorm, TNorm, and ZTNorm (or TZNorm). We use NIST 2004 data for the normalization. We select the normalization technique per system that gives the best performance. Table 10 shows the results. Score normalization gives a small but consistent improvement in the performance and does not affect the ranking of different approaches. Overall, the performance on NIST 2005 SRE improves from 7.03% to 5.71% (19% improvement) and the performance on NIST 2006 SRE improves from 5.12% to 3.83% (25% improvement). The improvement on NIST 2005 SRE is smaller than 2006 SRE because the choice of background and score normalization data (NIST 2004 SRE) is more matched to 2006 SRE.

Table 10 Score normalization on all the NAP results

System (with NAP)		SRE05		SRE06	
		%EER	DCF	%EER	DCF
1s1c		7.03	0.247	5.12	0.245
1s3c	Independent	5.99	0.219	4.69	0.230
	Joint	6.36	0.236	4.86	0.238
	Joint + separate class modeling	6.35	0.219	4.53	0.219
3s1c		7.07	0.253	5.29	0.246
3s3c	Independent	5.62	0.210	4.05	0.208
	Joint	6.27	0.214	4.04	0.196
	Joint + separate class modeling	5.71	0.185	3.83	0.192

6. Summary

We have presented a simplified and improved polynomial feature-based SVM system. The baseline system was proposed in [4] and has been used in NIST evaluations. We investigated the use of state-based and phone-based feature extensions where we used three states and three broad phone classes – vowels+diphthongs (VD), glides+nasals (GN) and obstruents (OB). We observed that state-based features are not useful on their own and phone-based features do improve performance. In combination, the joint state- and phone-based features improve performance relative to obtained with single-state phone classes.

We showed that independent modeling of phone classes and score-level combination performs significantly better than joint modeling. Comparison of results for phone-based features with single- and three-state features shows an interesting trend. First, state-based features give the least improvement for OBs. Second, these features give the most improvement for GNs. Finally, VDs provide the best performance among all phone categories.

Further, we applied NAP on different feature sets. Results show that the performance of joint NAP over all phone classes is similar to independent NAP per phone class with score-level combination, which is interesting for two reasons. First, phone-based features performed better when modeled independently so we hypothesized that the improvement would be consistent after NAP. However, the results showed that joint NAP takes advantage of the interclass correlations. This can be seen in the improvement in performance using GN features after joint NAP.

Joint NAP simplifies the system. It eliminates the computation of a separate NAP per-phone class, and seems to be more compact because it uses the same number of NAP dimensions as those for independent NAP.

Finally, we tested our hypothesis about the advantage of independent modeling of phone classes. We use the features after joint NAP and model them independently (as in the system without NAP). The score-level combination of the phone-based systems gives the best performance. Overall, we obtain about 25% relative improvement in performance by incorporating phone and state information.

7. Future Work

This work can be expanded in many ways. First, phones can be grouped more efficiently and in more than three categories. Second, the importance of states within each phone class needs to be investigated. In addition, some way of selecting important features should simplify the feature set and subsequent modeling. Finally, the idea of joint NAP followed by separate modeling and score-level combination can be applied to different feature sets.

8. Acknowledgments

The author thanks STAR Lab colleagues for helpful suggestions. He specifically thanks Andreas Stolcke for providing ASR transcripts, Luciana Ferrer for sharing ideas, and Nicolas Scheffer suggesting for SVDLIBC.

This work was funded in part by a Department of Defense KDD Award via NSF IIS-0544682, as well as through a development contract with Sandia National Laboratories. The views herein are those of the authors and do not necessarily represent the views of the funding agencies.

9. References

[1] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker Verification Using Adapted Mixture Models," *Digital Signal Processing*, vol. 10, pp. 181-202, 2000.

[2] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," presented at ICASSP, Orlando, 2002.

[3] W. M. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308-311, 2006.

[4] S. Kajarekar, "Four Weightings and a Fusion: A Cepstral-SVM System for Speaker Recognition," presented at ASRU, San Juan, 2005.

[5] S. S. Kajarekar, "Analysis of variability in speech with applications to speech and speaker recognition," in *Electrical Engineering*, vol. Ph. D. Portland, OR: OGI School of Science and Engineering at OHSU, 2002.

[6] A. Solomonoff, C. Quillen, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," presented at ICASSP, Philadelphia, USA, 2005.

[7] M. A. Przybocki, A. F. Martin, and A. N. Le, "NIST Speaker Recognition Evaluations Utilizing the Mixer Corpora—2004, 2005, 2006," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, pp. 1951-1959, 2007.

[8] NIST, "http://www.nist.gov/speech/tests/spk/index.htm."

[9] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification Systems," *Digital Signal Processing*, vol. 10, pp. 42-54, 2000.