ITERATED CLASS-SPECIFIC SUBSPACES FOR SPEAKER-DEPENDENT PHONEME CLASSIFICATION

Paul M. Baggenstoss

Naval Undersea Warfare Center
Newport RI, 02841, USA
phone: (+001) 401-832-8240
email: p.m.baggenstoss@ieee.org
web: www.nuwc.navy.mil/npt/csf/index.html
This work was supported by Office of Naval Research ONR 321 US

ABSTRACT

The features based on the MEL cepstrum have long dominated probabilistic methods in automatic speech recognition (ASR). This feature set has evolved to maximize general ASR performance within a Bayesian classifier framework using a common feature space. Now, however, with the advent of the PDF projection theorem (PPT) and the class-specific method (CSM), it is possible to design features separately for each phoneme and compare log-likelihood values fairly across various feature sets. In this paper, class-dependent features are found by optimizing a set of frequency-band functions for projection of the spectral vectors, analogous to the MEL frequency band functions, individually for each class. Using this method, we show significant improvement over standard MEL cepstrum methods in speaker and phoneme specific recognition.

1. INTRODUCTION

The MEL cepstrum features [1] and its derivatives have long been the staple of automatic speech recogniton (ASR) systems. One may write the MEL cepstrum features as

$$\mathbf{z} = \mathrm{DCT}(\log(\mathbf{A}'\mathbf{y})), \tag{1}$$

where vector $\mathbf{y}$ is the length-$N/2+1$ spectral vector, the magnitude-squared DFT output and the columns of $\mathbf{A}$ are the MEL band functions [1]. In typical speech applications, only the first L elements of $\mathbf{z}$ are preserved. Because the logarithm and the discrete cosine transform (DCT) may be considered a feature conditioning step which results in more Gaussian-like and independent features, we may concentrate our attention on the matrix multiplication

$$\mathbf{w} = \mathbf{A}'\mathbf{y}. \tag{2}$$

The key operation here is dimension reduction by linear projection onto a lower-dimensional space. Now, with the introduction of the class-specific method (CSM) and the PDF projection theorem (PPT) [2], one is free to explore class-dependent features within the rigid framework of Bayesian classification. Some work has been done in class-dependent features [3],[4] however existing approaches are only able to use different features through the use of compentation factors to make likelihood comparisons fair. Such approaches work if the class-dependent feaure transformations are restricted to certain limited sets. Both methods fall short of the potential of the PPT which makes no restriction on the type of feature transformations available to each phoneme. Under CSM, the "common feature space" is the time-series (raw data) itself. Feature PDFs, evaluated on different feature spaces are projected back to the raw data space where the likelihood comparison is done. Besides its generality, the CSM paradigm has many additional advantages as well. For example there is a quantitative class-dependent measure to optimize that allows the design of the class-dependent features in isolation, without regard to the other classes.

2. CLASS-SPECIFIC APPROACH

When applying CSM, one must find class-dependent signal processing to produce features that characterize each class or sub-class. We seek an automatic means of optimizing the matrix $\mathbf{A}$ for a given subclass. We first review CSM.

2.1 Class-Specific Method (CSM)

Let there be M classes among which we would like to classify. The class-specific classifier, based on the PPT, is given by

$$\arg\max_m p_p(\mathbf{x}|H_m),$$

where $p_p(\mathbf{x}|H_m)$ is the *projected* PDF (projected from the feature space to the raw data space). The projected PDF is given by

$$p_p(\mathbf{x}|H_m) = J_m(\mathbf{x}, \mathbf{A}_m, H_{0,m})\ \hat{p}(\mathbf{z}_m|H_m), \tag{3}$$

where $\hat{p}(\mathbf{z}_m|H_m)$ is the feature PDF estimate (estimated from training data) and the J-function is given by

$$J_m(\mathbf{x}, \mathbf{A}_m, H_{0,m}) = \frac{p(\mathbf{x}|H_{0,m})}{p(\mathbf{z}_m|H_{0,m})}, \tag{4}$$

and $H_{0,m}$ are class-dependent reference hypotheses. The class-dependent features $\mathbf{z}_m$ are computed from the spectral vector $\mathbf{y}$ through the class-dependent subspace matrices $\mathbf{A}_m$, as

$$\mathbf{z}_m = C(\mathbf{A}_m'\mathbf{y}), \tag{5}$$

where C is the feature conditioning transformation. Note that the J-function is a fixed function of $\mathbf{x}$ precicely defined by the feature transformation from $\mathbf{x}$ to $\mathbf{z}$ and the reference hypotheses $H_{0,m}$. It is the "correction term" that allows feature PDFs from various feature spaces to be compared fairly because the resulting log-likelihood function is a PDF on the raw data space $\mathbf{x}$. The J-function is a generalization of the determinant of the Jacobian matrix in the case of a 1:1 transformation. The PPT guarantees that $p_p(\mathbf{x}|H_m)$ given by (3) is a PDF, so it integrates to 1 over $\mathbf{x}$ *regardless* of the reference hypothesis $H_{0,m}$ or the feature transformation producing $\mathbf{z}_m$ from $\mathbf{x}$. It is up to the designer to choose $H_{0,m}$ and $\mathbf{A}_m$ to make $p_p(\mathbf{x}|H_m)$ as good an estimate of $p(\mathbf{x}|H_m)$ as possible. The designer is guided by the principle that if $\mathbf{z}_m$ is a suffient statistic for H_m vs. $H_{0,m}$, then $p_p(\mathbf{x}|H_m)$ will equal $p(\mathbf{x}|H_m)$ (provided $\hat{p}(\mathbf{z}_m|H_m)$ is a good estimate). We can also think of it as a way of embedding a low-dimensional PDF within a high-dimensional PDF.

We have good reason, as we shall see, to use a common reference hypothesis, H_0, which simplifies the classifier to

$$\arg\max_m J_m(\mathbf{x}, \mathbf{A}_m, H_0)\ p(\mathbf{z}_m|H_m) \tag{6}$$

where the J-function $J_m(\mathbf{x})$ now depends only on $\mathbf{A}_m$. Note that in contrast to other class-dependent schemes using pairwise or tree tests, CSM is a Bayesian classifier and has the promise CSM of providing a "drop-in" replacement to the MEL-cepstrum based feature processors in existing ASR systems.

2.2 Finding a class-specific subspace

We are interested in adapting the matrix $\mathbf{A}$ to an individual class. We propose the strategy of selecting $\mathbf{A}_m$ to maximize the total log-likelihood of the training data using the projected PDF. Let

$$L(\mathbf{x}^1, \mathbf{x}^2 \ldots \mathbf{x}^K; \mathbf{A}_m) = \sum_{i=1}^{K} \log p_p(\mathbf{x}^i|H_m) \tag{7}$$

where K is the number of training vectors. If we expand $p_p(\mathbf{x}|H_m)$,

$$p_p(\mathbf{x}|H_m) = \left[\frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_m|H_0)}\right]\ \hat{p}(\mathbf{z}_m|H_m),$$

where H_0 is the independent Gaussian noise hypothesis, we see that the term $p(\mathbf{x}|H_0)$ is independent of $\mathbf{A}_m$. Thus, to maximize L, we need to maximize the average value of

$$\log \hat{p}(\mathbf{z}_m|H_m) - \log p(\mathbf{z}_m|H_0). \tag{8}$$

Our approach is to assume that the first term in (8) is only weakly dependent on $\mathbf{A}_m$ and concentrate on the second term. Given the simplicity of the reference hypothesis H_0, the second term $p(\mathbf{z}_m|H_0)$ can be known, either in analytic form or in an accurate analytic approximation [5]. Thus, it is easy to analyze its behavior as $\mathbf{A}_m$ changes. We have obtained the first derivatives of $\log p(\mathbf{z}_m|H_0)$ with respect to each element of $\mathbf{A}_m$. We proceed, then by ignoring the term $\hat{p}(\mathbf{z}_m|H_m)$ and maximizing the function

$$Q(\mathbf{x}^1, \mathbf{x}^2 \ldots \mathbf{x}^K; \mathbf{A}_m) = -\sum_{i=1}^{K} \log p(\mathbf{z}_m^i|H_0). \tag{9}$$

The change in $\hat{p}(\mathbf{z}_m|H_m)$ can be minimized as $\mathbf{A}_m$ is changed by insisting on an orthonormal form for $\mathbf{A}_m$. Thus, by maximizing L (7) under the restriction that $\mathbf{A}_m$ is orthonormal, we approximately maximize L. We apply the following constraints to $\mathbf{A}_m$:

- **Orthonormality**. The columns of $\mathbf{A}_m$ are an orthonormal set of vectors. We use a orthonormality under the inner product

$$< \mathbf{x}, \mathbf{y} >= \sum_{i=0}^{N/2} \varepsilon_i x_i y_i,$$

  where ε_i has the value 2 except for the end bins (0 and $N/2$) where it has value 1. Ortho-normality under this inner product means that the spectral vectors will be orthonormal if extended to the full N bins. Use of orthonormality helps to stabilize the term $\hat{p}(\mathbf{z}_m|H_m)$ as $\mathbf{A}_m$ is varied.
- **Energy sufficiency**. The energy sufficiency constraint means that the total energy in $\mathbf{x}$,

$$E = \sum_{i=1}^{N} x_i^2$$

  can be derived from the features. Energy sufficiency is important in the context of floating reference hypotheses [2]. In order that the classifier result is scale invariant, we need energy sufficiency. With energy sufficiency, the term

$$\frac{p(\mathbf{x}|H_0)}{p(\mathbf{z}_m|H_0)}$$

  will be independent of the variance used on the H_0 reference hypothesis. Note that $E = \mathbf{e}_1'\mathbf{y}/N$, where $\mathbf{e}_1 = [1, 2, 2, 2\ldots, 2, 1]'$, which is composed of the number of degrees of freedom in each frequency bin. Thus, energy sufficiency means that the column space of $\mathbf{A}_m$ needs to contain the vector $\mathbf{e}_1$.

2.2.1 Class-specific iterated subspace (CSIS)

Since we would like the feature set created by projecting onto the columns of $\mathbf{A}$ to characterize the statistical variations within the class, a natural first step is to use principal component analysis (PCA). To do this, we arrange the spectral vectors from the training set into a matrix

$$\mathbf{X} = [\mathbf{y}^1 \mathbf{y}^2 \cdots \mathbf{y}^K],$$

where K is the number of training vectors. To meet the energy sufficiency constraint, we fix the first column of $\mathbf{A}$ to be the normalized $\mathbf{e}_1$

$$\tilde{\mathbf{e}_1} = \frac{\mathbf{e}_1}{\|\mathbf{e}_1\|}.$$

To find the best linear subspace orthogonal to $\mathbf{e}_1$, we first orthogonalize the columns of $\mathbf{X}$ to $\mathbf{e}_1$ $\mathbf{X}_n = \mathbf{X} - \tilde{\mathbf{e}}_1(\tilde{\mathbf{e}_1}'\mathbf{X})$. Let $\mathbf{U}$ be the largest P singular vectors of $\mathbf{X}_n$, or equivalently the largest P eigenvectors of $\mathbf{X}_n\mathbf{X}_n'$. We then set $\mathbf{A} = [\tilde{\mathbf{e}}_1\mathbf{U}]$. We then proceed to maximize (9) using an iterative approach. We use the term class-specific iterated subspace (CSIS) to refer to the columns of $\mathbf{A}_m$ obtained in this way.

3. TECHNICAL APPROACH

3.1 Data Set

We used the TIMIT [6] data set as a source of phonemes, drawing all of our data from the "training" portion. TIMIT consists of sampled time-series (in 16 kHz .wav files) of scripted sentences read by a wide variety of speakers and includes index tables that point to start and stop samples of each spoken phoneme in the text. There are 61 phonemes in the database, having a 1 to 4 character code. We use the term *dataclass* to represent the collection of all the phonemes of a given type from a given speaker. The average number of samples (utterences) of a given speaker/phoneme combination is about 10 and ranges from 1 up to about 30 for some of the most common phonemes. We used speaker/phoneme combinations with no fewer than 10 samples.

3.2 Cross-Validation

In all of our classification experiments, the utterences of a given speaker/phoneme were divided into two sets, even (utterences 2,4,6 ...) and odd (utterences 1,3,5...). We conducted two sub-experiments, training on even, testing on odd, then training on odd, testing on even. We reported the sum of the classification counts from the two experiments.

3.3 Processing

We now describe the processing for the features of the MEL frequency cepstral coefficient (MFCC) classifier and CSIS. In order to concentrate on the basic dimension reduction step (equation 2), the simplest possible processing and PDF modeling was used. Each step in the processing is described below, in the order in which it is processed.

3.3.1 Resampling

We pre-processed all TIMIT .wav files by resampling from 16 kHz down to 12 kHz. Phoneme endpoints were correspondingly converted and used to select data from the 12 kHz time-series.

3.3.2 Truncation

The phoneme data was truncated to a multiple of 384 samples by truncating off the end. Those phoneme events that were below 384 samples at 12 kHz were dropped. Doing this allowed us to use FFT sizes of 48, 64, 96, 128, or 192 samples, which are all factors of 384.

3.3.3 FFT processing

We computed non-overlapped unshaded (rectangular window function) FFTs resulting in a sequence of magnitude-squared FFT spectral vectors of length $N/2+1$, where N is the FFT size. The number of FFTs in the sequence depended on how many non-overlapped FFTs fit within the truncated phoneme utterance.

3.3.4 Spectral normalization

Spectral vectors were normalized after FFT processing. For non-speaker-dependent (MEL cepstrum) features, the spectral vectors were normalized by the average spectrum of all available data.

For CSIS (speaker-dependent) features, the spectral values for each speaker/phoneme combination were normalized by the average spectrum for that speaker/phoneme. In classification experiments the average spectrum was computed from the training data to avoid issues of data separation.

3.3.5 Subspace Projection (Matrix Multiplication)

Next, the spectral vectors, denoted by $\mathbf{y}$, were projected onto a lower dimensional subspace by a matrix as in (2) resulting in feature vectors, denoted by $\mathbf{w}$.

For MFCC, the columns of $\mathbf{A}$ were MEL frequency band functions. The number of columns in matrix $\mathbf{A}$ was N_c+2 including the zero and Nyquist half-bands and the number of DCT coefficients was L.

For CSIS, $\mathbf{A}$ was an orthonormal matrix determined from the optimization algorithm. For CSIS, the number of columns of $\mathbf{A}$ was $P+1$ where P is the number of basis functions in addition to the first column $\tilde{\mathbf{e}}_1$.

3.3.6 Feature Conditioning

From a statistical point of view, feature conditioning has no effect on the information content of the features. It does, however, make probability density function (PDF) estimation easier if the resulting features are approximately independent and Gaussian. For MFCC, the features were conditioned by taking the log and DCT as in (1), then retaining the first L coefficients. For CSIS, features were conditioned first by dividing features 2 through $P+1$ by the first feature. This effectively normalizes the features since the first feature, being a projection onto $\mathbf{e}_1$, is a power estimate for the segment. Lastly, the log of the first feature is taken. Mathematically, we have for CSIS

$$\mathbf{w} = \mathbf{A}'\mathbf{y},$$

$$z_1 = \log(w_1),$$

$$z_i = w_i/w_1,\ i = 2,3,\ldots P+1,$$

where w_1 is the first element of $\mathbf{w}$ and equals $\tilde{\mathbf{e}}_1'\mathbf{y}$.

3.3.7 J-function calculation

J-function contributions must be included for FFT magnitude-squared, spectral normalization, matrix multiplication, and feature conditioning. See [7] for details of these class-specific modules.

3.3.8 PDF modeling and Classification

We intentionally used an over-simplified process for PDF estimation and classification. One reason is that the use of overlapped window processing, which is common in ASR, is difficult to achieve using CSM due to the statistical dependence of overlapped segments and the difficulty in deriving the denominator term in (4). To deal with segment-to-data registration issues, on-the fly segmentation, such as that used in multi-resolution HMM [8], is more appropriate for CSM. The use of rectangularly-shaded FFTs and simple likelihood summation is a necessary compromise allows MFCC and CSM to be meanigfully compared. Furthermore, by doing this, we made it easier to interpret the results because differences in performance could only be attributed to the differences in the choice of matrix $\mathbf{A}$.

For PDF estimation, we used a simple multivariate Gaussian PDF model, or equivalently a Gaussian mixture model (GMM) with a single mixture component. We assume independence between the members of the sequence within a given utterence, thus disregarding the time ordering. The log-likelihood value of a sample was obtained by evaluating the total log-likelihood of the feature sequence from the phoneme utterance. The reason we used such simplified processing and PDF models was to concentrate our discussion on the features themselves. Classification was accomplished by maximization of log-likelihood across class models. For CSS and CSIS, we added the log J-function value to the log-likelihood value of the GMM [2], implementing (6) in the log domain.

4. EXPERIMENTAL RESULTS

4.1 Validation of Assumptions

An important experiment to perform is to validate the assumption used in section 2.2, that maximizing L (equation 7) can be achieved by maximizing Q in equation (9). Although space does not permit presenting the results, we have obtained overwhelming empirical evidence that the second term in (8) does in fact dominate.

4.2 Choice of FFT size and model order

The CSIS approach is parameterized by two parameters, the FFT size N, and the model order P. The MFCC method is parameterized by the FFT size N, the number of MEL bands N_c, and the number of DCT coefficients L. We chose to use the same value of N for MFCC and CSIS. This ensured that the only significant difference between MFCC and CSIS would be the matrix $\mathbf{A}$ which is chosen as a function of class thanks to the PPT. Feature conditioning is also different but is not expected to contribute greatly to performance differences. For fair comparison, we operated the MFCC at the parameter settings that produced the fewest total errors, which turned out to be $N = 96$, $N_c = 25$, and $L = 12$. The values of $N_c = 25$, and $L = 12$ are commonly used in speech applications, but $N = 96$, being 8 milliseconds, is significantly smaller than usual. The reason for this may be the fact that we used rectangular shading which has the same effective number of samples as a 16 ms window that is more common. In any case, this combination gave the best results for MFCC of all those we tried which included $N = \{48, 96, 128, 192, 384\}$, $L = \{6, 8, 10, 12, 14\}$, $N_c = \{16, 20, 25\}$. For CSIS, we are left with deciding on the model order P. Refer to figure 1. In which we see the total

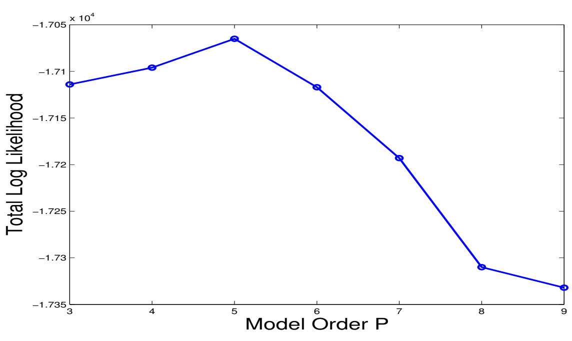

Figure 1: Total log-likelihood (with even-odd cross-validation) as a function of P for speaker `MGRL0` phoneme "N" with CSIS.

log-likelihood L of speaker `MGRL0`,phoneme "N", as a function of P. Even-odd cross-validation is used (section 3.2). Note that the likelihood increases up to $P = 5$ then exhibits a steep decline. This suggests that a dimension-5 subspace is optimal to represent this speaker/phoneme combination. For the individual speaker experiments, we chose model order for each speaker/phoneme combination in the same way. When we expanded the data to all male speakers of "N", we attained a peak at $P = 8$. This indicates that an increase in subspace dimension is required for multiple speakers.

Experiment	MFCC	CSIS(5)	CSIS
IY vs. EH	45	49	47
AE vs. EH	46	43	40
R vs. L	64	57	52
AX vs. AXR	84	68	66
IX vs. IH	85	86	80
N vs. M	82	48	47
DCL vs. TCL	96	84	86
Total	502	435	418

Table 1: Error metric E_c as a function of phoneme pair and classifier type. CSIS(5) indicates that model order P is fixed at $P = 5$ while for CSIS, it is chosen separately per class.

4.3 Classification Experiments

We conducted both *two-phoneme* experiments and *single-speaker* experiments which will be described separately. All of our experiments used strict separation between training and testing data (section 3.2). Our primary performance metric, which we denoted by E_c, measured all off-diagonal elements in the confusion matrix. That is, any error, either identifying the wrong phoneme or speaker was counted.

4.3.1 Two-Phoneme Experiments.

The two-phoneme experiments were designed to test the ability to distinguish speakers of a given phoneme as well as classify two phonemes in a limited multi-speaker environment. We selected fourteen phonemes arranged into seven experiments, each involving two phonemes. For each phoneme, we chose a set of from four to seven individual speakers of the same sex. We selected phoneme/speaker combinations that had large numbers of utterences per speaker - a minimum of ten utterences per speaker and an average of about 12 utterences per speaker/phoneme combination. Thus, there were about 60 utterences per phoneme.

The results are tabulated in table 1. First, CSIS, with P chosen separately for each class performed generally better than CSIS(5) which uses model order fixed at $P = 5$. This indicates that individually optimized model order is better. The fact that the model orders were determined individually without regard to other classes validates is an important observation. In comparison to MFCC, CSIS achieved a lower E_c in all experiments except one where it was comparable. On the whole, MFCC produced 20 percent more errors. For the difficult N vs. M problem, it produced 74 percent more errors. Although there is insufficient space for much detail, we note that MFCC produced fewer inter-phoneme errors.

4.3.2 Single-Speaker Experiments.

The single-speaker experiments were designed to test the ability to distinguish phonemes of a given speaker. In each of the seventeen single-speaker experiments, we gathered data from a single speaker and between four and seven phonemes into one classification experiment and measured E_c. The results are tabulated in table 2. The results reflect the results

Speaker	MFCC	CSIS(5)	CSIS
fmem0	19	20	18
fceg0	36	32	20
mkag0	37	25	21
fapb0	32	28	29
mcxm0	38	23	22
mmea0	23	21	22
fdaw0	30	23	24
mgrl0	22	16	15
mkdd0	27	21	19
msat1	23	20	19
mbma1	43	26	26
mprk0	23	20	19
fklh0	25	22	21
mjma0	24	21	19
mbth0	18	16	14
mbcg0	26	23	23
mmlm0	29	28	30
Total	475	385	361

Table 2: Comparison of MFCC and CSIS in single-speaker experiments using error metric E_c. CSIS(5) indicates CSIS with model order fixed at $P = 5$.

of the two-phoneme experiments. CSIS performed better than MFCC except in one experiment where it was comparable. Total error is 31 percent higher for MFCC. Furthermore, CSIS performs better when model order is individually selected (section 4.2). This is significant because in addition to matrix $\mathbf{A}$ being a function of class, the feature dimension is also a function of class.

4.4 Discussion of Results

We can draw some meaningful conclusions from the experiments. First, we see that both in discriminating phonemes of a given speaker and in discriminating speakers of a given phoneme, CSIS is consistently better than MFCC. Although we had insufficient space for details, when we measured the inter-phoneme errors in the two-phoneme experiments (disregarding speaker identity), MFCC did generally better. The reason may lie in the shrinking of the linear subspace as we restrict ourselves to a single speaker/single phoneme. When the subspace is limited, CSIS may be able to find a better statistical model of the distributiuon. A second piece of evidence that supports this is the fact that the highest improvement of CSIS over MFCC was obtained in the experiment "N-vs-M" which is one of the most difficult problems in ASR, an indication that CSIS produces a better PDF estimate at the center of the distributions. Thus, when classes are more close to each other, i.e. overlapped, the better PDF estimate will be more important, because the optimal decision boundary is given by the true likelihood ratio. However, since MFCC has evolved for phoneme discrimination, it performs better than CSIS in the inter-phomeme areas. When two phonemes are very similar, discrimination occurs "near the peak" where CSIS performs better.

Future work should determine how can the strengths of both CSIS and MFCC be best utilized. The evidence we provided suggests that the most promising approach for applying CSIS to multi-speaker experiments may lie in the ability to cluster speakers into like-sounding groups, which can be represented by separate low-dimensional CSIS models.

REFERENCES

[1] J. W. Picone, "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, vol. 81, no. 9, pp. 1215–1247, 1993.

[2] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans Signal Processing*, pp. 672–685, March 2003.

[3] H. Watanabe and S. Katagiri, "Hmm speech recognizer based on discriminative metric design," in *Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP97)*, vol. 4, 1997.

[4] M. K. Omar and M. Hasegawa-Johnson, "Strong-sense class-dependent features for statistical recognition," in *Proc. IEEE Workshop on Statistical Signal Processing*, vol. 4, pp. 490–493, 2003.

[5] S. M. Kay, A. H. Nuttall, and P. M. Baggenstoss, "Multidimensional probability density function approximation for detection, classification and model order selection," *IEEE Trans. Signal Processing*, pp. 2240–2252, Oct 2001.

[6] J. S. Garofolo, "Timit acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993.

[7] P. M. Baggenstoss, "The class-specific classifier: Avoiding the curse of dimensionality (tutorial)," *IEEE Aerospace and Electronic Systems Magazine, special Tutorial addendum*, vol. 19, pp. 37–52, January 2002.

[8] P. M. Baggenstoss, "A multi-resolution hidden markov model using class-specific features," *EUSIPCO 2008, Lausanne, August 25-29*, 2008.