Noname manuscript No.
(will be inserted by the editor)

Feature Selection for Robust Automatic Speech Recognition: A Temporal Offset Approach

Ludovic Trottier · Philippe Giguère · Brahim Chaib-draa

Received: September 29, 2014 / Accepted: March 4, 2015

Abstract Automatic speech recognition relies on extracting features at fixed intervals. In order to enhance these features with dynamical (delta) components, discrete derivatives are usually computed and added as features. However, derivative operations tend to be susceptible to noise. Our proposed method alleviates this problem by replacing these derivatives with nearby features selected on a per-frequency basis. In particular, we noted that, at low frequency, consecutive samples are highly correlated and more information can be gathered by looking at features farther away in time. We thus propose a strategy to perform this frequency-based selection and evaluate it on the Aurora 2 continuous-digits and connected-digits tasks using MFCC, PLPCC and LPCC standard features. The results of our experimentations show that our strategy achieved an average relative improvement of 32.10% in accuracy, with most gains in very noisy environments where the traditional delta features have low recognition rates.

Keywords automatic speech recognition, delta features, feature extraction, noise robustness.

L. Trottier
Department of Computer Science and Software Engineering,
Laval University, Québec, Canada
E-mail: ludovic.trottier.1@ulaval.ca

P. Giguère
Department of Computer Science and Software Engineering,
Laval University, Québec, Canada
E-mail: philippe.giguere@ift.ulaval.ca

B. Chaib-draa
Department of Computer Science and Software Engineering,
Laval University, Québec, Canada
E-mail: chaib@ift.ulaval.ca

1 Introduction

Automatic speech recognition (ASR) is the transcription of spoken utterances into text. A system that performs ASR tasks takes an audio signal as input and classifies it into a series of words. In order to help the system accomplish its task, it is essential to process the signal and provide reliable features. The three most frequently used features for ASR are the Mel frequency cepstral coefficients (MFCC), the perceptual linear predictive cepstral coefficients (PLPCC) and the linear predictive cepstral coefficients (LPCC) (see [Shrawankar and Thakare, 2013] for a review). These filter bank analysis extraction methods use various transformations, such as the Fourier transform, to convert a signal into a series of static vectors called *feature frames*. The coefficients in a feature frame are usually ordered from low-frequency to high-frequency and this observation will play a central role in our approach.

Classical feature extraction methods enhance each feature frame with dynamical components by applying discrete time derivatives. The idea of the concatenation of first- and second-order derivatives, dubbed *delta* features, was proposed (in a similar form) as a way to improve the spectral dynamics of static features [Furui, 1986]. Even though it has been evaluated that the delta features achieve great results [Zheng et al., 2001], it is known from signal processing theories that the derivative of a noisy signal amplifies the noise, thus reducing the quality of the extracted information [Oppenheim et al., 1999]. This can be particularly detrimental for situations where the presence of noise adversely affects the recognition, such as when driving a car [Lockwood and Boudy, 1992].

We have proposed, in a preliminary approach, that the discrete time derivatives could be replaced with a mere concatenation of adjacent (in time) coefficients based on frequency [Trottier et al., 2014]. This approach will be referred to as Temporal Feature Selection (TFS). The idea that the concatenation of dynamical features should take into account the frequency of the components comes from the fact that it is essential to model inter-frame dependencies for speech utterances. Signal processing theories suggest that the way the information varies in a signal depends on frequency [Oppenheim et al., 1999]. For example, implosive consonants will result in fast, high-frequency features, while vowels will produce slow-changing, lower-frequency features. It thus appears that frequency is a good metric of the variation of the information in a signal and could be used to improve the dynamical features.

In this paper, we extend our TFS method by providing a simple framework to learn them. Our framework uses the variance of the difference between feature frames as a way to identify the time-delay at which features are sufficiently decorrelated. We show experimentally that our dynamical features improve the accuracy over the classical delta features on the Aurora 2 [Pearce et al., 2000] database.

The rest of the paper is organized as follows. Section 2 describes related approaches, Section 3 presents the TFS method, Section 4 details the experimentations, Section 5 contains a discussion of the results and Section 6 concludes this work.

2 Related Work

To overcome the aforementioned problem of delta features, recent attempts have been investigated. Thus, the delta-spectral cepstral coefficients (DSCC) were proposed in replacement of the delta features in order to increase the robustness to additive noise [Kumar et al., 2011]. Also, the distributed discrete cosine transform has been proposed to replace the classical discrete cosine transform prior to the computation of delta features [Hossan et al., 2010]. Finally, instead of concatenating the static and delta features, a weighted sum combining them has been proposed [Weng et al., 2010]. However, the main drawback of all these methods is that they still make use of derivatives and are prone to being corrupted by noise.

Additional methods have tried to improve speech features in various ways. In the context of deep learning, splicing followed by decorrelation and dimensionality reduction has been used to enhance the input of deep neural networks (DNNs) [Rath et al., 2013]. Splicing consists in concatenating all feature frames (with delta features) in a *context window* of size c around each frame [Bahl et al., 1994]. This approach is however too conservative since the majority of the concatenated features are either redundant or non-informative. The other disadvantage of splicing is that the dimensionality of the neural network input layer is very large which makes the parameter inference unnecessarily harder. Moreover, the benefits of depth in DNNs has been investigated and it was concluded that additional layers allow more discriminative and invariant features to be learned [Yu et al., 2013]. While we acknowledge that deep learning is a promising avenue for feature extraction in ASR, we argue that better feature engineering methods could facilitate the DNN learning process.

When adjacent feature frames are concatenated (as in splicing), or when higher-order delta features are used (e.g. up to the third), the inputs' dimensionality can be too large. Some researchers have proposed improving speech features with dimensionality reduction approaches. Principal Component Analysis (PCA) has been applied to project the data while retaining maximum variance [Jolliffe, 1986]. To obtain subspaces that discriminate better between the classes, Linear Discriminant Analysis (LDA) has been proposed [Fukunaga, 1990]. The LDA criterion was further improved by using the true class covariance matrices as in Heteroscedastic Discriminant Analysis (HDA) [Saon et al., 2000], or by employing Heteroscedastic LDA (HLDA) [Kumar and Andreou, 1998] when the classes have the same means but different covariances. Even though HLDA usually out-performs LDA, it is more computationally expensive in time and memory space [Kumar and Andreou, 1998]. While HLDA appears to be an essential tool for speech feature extraction, we argue that dimensionality reduction may be avoided by using a more expert strategy when gathering the signal dynamics.

In the context of linear feature transformations unrelated to dimensionality reduction, Maximum Likelihood Linear Transform (MLLT) [Gopinath, 1998] and Global Semi-tied Covariance (GSC) [Gales, 1999] have been proposed as decorrelation approaches. MLLT finds a linear transformation that maximizes the likelihood of the observations under isotropic Gaussian densities. On the other hand, GSC decorrelates the features by using the eigen decomposition of each state-specific covariance. Moreover, feature-space Maximum Likelihood Linear Regression (fMLLR) [Leggetter and Woodland, 1995] and Constrained MLLR (CMLLR) [Gales, 1998] have been proposed as linear transformation approaches for speaker adaptation. fMLLR directly modifies the parameters of the Gaussian densities while CMLLR changes the features themselves. Even though these approaches are essential feature selection techniques, they do not address the problem of modeling the dynamics of speech.

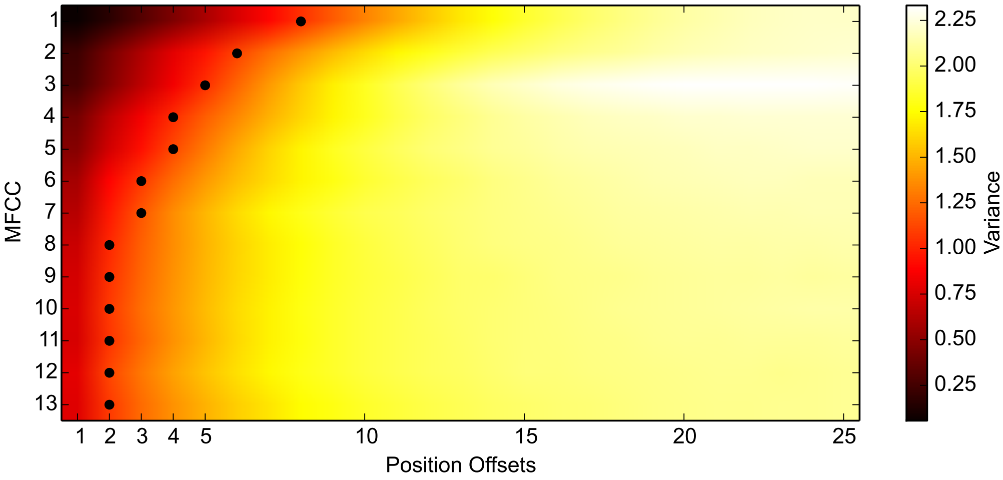


Fig. 1 Variance of the difference between a frame and its neighbors for MFCC features on the Aurora 2 [Pearce et al., 2000] training dataset (best seen in colors). The coefficients are ordered from low frequency (1) to high frequency (13) for visual convenience (the proposed method does not require a specific ordering). The color refers to the variance of the difference Σ^M, where M was limited to 25 to reduce the computational burden.

3 Temporal Feature Selection

3.1 Definition

Let $\Phi^{(n)} = \left(\boldsymbol{\phi}^{(n)}_{:,1} \dots \boldsymbol{\phi}^{(n)}_{:,T_n}\right)$, $n = 1 \dots N$, be a $D \times T_n$ matrix of D-dimensional static features. N is the total number of utterances and T_n denotes the number of frames extracted from utterance n. For example, $\Phi^{(n)}$ could represent spectrograms as well as MFCC. We denote the column vector $\boldsymbol{\phi}^{(n)}_{:,t}$ as the feature frame at position t. The classical method of computing the delta features uses the following equations:

$$\Delta\boldsymbol{\phi}^{(n)}_{:,t} = \frac{\sum_{k=1}^{K} k\left(\boldsymbol{\phi}^{(n)}_{:,t+k} - \boldsymbol{\phi}^{(n)}_{:,t-k}\right)}{2\sum_{k=1}^{K} k^2}\,, \tag{1}$$

$$\Delta\Delta\boldsymbol{\phi}^{(n)}_{:,t} = \frac{\sum_{k=1}^{K} k\left(\Delta\boldsymbol{\phi}^{(n)}_{:,t+k} - \Delta\boldsymbol{\phi}^{(n)}_{:,t-k}\right)}{2\sum_{k=1}^{K} k^2}\,, \tag{2}$$

where $K = 2$ is a typical value for the summation. Although this subtraction allows for the extraction of dynamical information about adjacent features, it is also susceptible to noise.

The TFS features are, in contrast, coefficients taken from adjacent feature frames based on the frame position offsets $\mathbf{z} = [z_1, \dots, z_D]$. We define them as:

$$\tau\boldsymbol{\phi}^{(n)}_{i,t} = \left(\boldsymbol{\phi}^{(n)}_{i,t+z_i}, \boldsymbol{\phi}^{(n)}_{i,t-z_i}\right), \tag{3}$$

where z_i is a strictly positive integer that depends on the frequency. $\mathbf{z}$ should try to select coefficients $\boldsymbol{\phi}$ that are dissimilar, but not too much. Too similar values do not increase the amount of information the feature frames carry, but increase its dimensionality, and this makes the speech recognition task harder. If the coefficients are too far apart, then their temporal correlation is meaningless.

3.2 Learning the TFS Features

We now present the proposed framework to learn the offsets $\mathbf{z}$. The method first computes the sample variance of the difference of neighboring feature frames. In other words, for each position t and utterance n, the difference between the feature frame $\boldsymbol{\phi}^{(n)}_{:,t}$ and its corresponding j^{th} neighbor $\boldsymbol{\phi}^{(n)}_{:,t+j}$ is computed. The variance of these differences is then calculated for $j \in \{1 \dots M\}$, where $M = \min\{T_1 \dots T_N\} - 1$. We define the matrix containing those values as:

$$\Sigma^M = \begin{bmatrix} \mathrm{Var}\left(\boldsymbol{\phi}^{(n)}_{1,t} - \boldsymbol{\phi}^{(n)}_{1,t+1}\right) & \dots & \mathrm{Var}\left(\boldsymbol{\phi}^{(n)}_{1,t} - \boldsymbol{\phi}^{(n)}_{1,t+M}\right) \\ \vdots & & \vdots \\ \mathrm{Var}\left(\boldsymbol{\phi}^{(n)}_{D,t} - \boldsymbol{\phi}^{(n)}_{D,t+1}\right) & \dots & \mathrm{Var}\left(\boldsymbol{\phi}^{(n)}_{D,t} - \boldsymbol{\phi}^{(n)}_{D,t+M}\right) \end{bmatrix} \tag{4}$$

where the variances are taken over all positions t and utterances n. The variance is then computed as follows:

$$\Sigma^M_{i,j} = \frac{1}{N^+_j} \sum_{n=1}^{N} \sum_{t=1}^{T_n - j} \left(\boldsymbol{\phi}^{(n)}_{i,t} - \boldsymbol{\phi}^{(n)}_{i,t+j} - \mu_{i,j}\right)^2, \tag{5}$$

where $\mu_{i,j}$ corresponds to the mean of the difference:

$$\mu_{i,j} = \frac{1}{N_j^+} \sum_{n=1}^{N} \sum_{t=1}^{T_n - j} \left(\boldsymbol{\phi}_{i,t}^{(n)} - \boldsymbol{\phi}_{i,t+j}^{(n)} \right) , \tag{6}$$

and N_j^+ is the total number of frames:

$$N_j^+ = \sum_{n=1}^{N} T_n - j \, . \tag{7}$$

The purpose of computing Σ^M is to find the frame position offsets $\mathbf{z}$. Using the parameter V_{thresh} as a variance threshold, $\mathbf{z}$ is computed using the following equation:

$$z_i = \arg\min_j \left| \Sigma_{i,j}^M - V_{thresh} \right| , \tag{8}$$

where V_{thresh} is a hyper-parameter to choose.

The frame position offsets $\mathbf{z}$ represented in Fig. 1 by the black dots are based on Eq. 8 for $V_{thresh} = 1$ and $M = 25$. In this example, $z_1 = 8$ and $z_{13} = 2$. This implies that the TFS features of $\boldsymbol{\phi}_{1,t}$ and $\boldsymbol{\phi}_{13,t}$ are $(\boldsymbol{\phi}_{1,t+8}, \boldsymbol{\phi}_{1,t-8})$ and $(\boldsymbol{\phi}_{13,t+2}, \boldsymbol{\phi}_{13,t-2})$.

What can be seen from this figure is that $\mathbf{z}$ depends on frequency. High frequency components have small offsets whereas low frequency components have large offsets. As explained in Section 1, more reliable dynamical information can be extracted from neighboring feature frames when frequency is taken into account. The relevant dynamical information of high frequency coefficients can only be extracted from nearly adjacent frames ($z_{13} = 2$). On the other hand, adjacent low frequency coefficients share most of their information and more time is needed to gather the relevant dynamics ($z_1 = 8$). Therefore, by using the variance of neighboring feature frames, $\mathbf{z}$ now incorporates the wanted characteristic of frequency dependency.

Once the configuration of $\mathbf{z}$ is inferred from the data, it remains fixed. A further modification of our approach is to compute the time offsets $\mathbf{z}$ for each utterance independently. In other words, the variances in Eq. 4 are computed over t only for each n separately. This variation of the proposed method will be evaluated in Section 4.

3.3 Hand-Designed Offsets $\mathbf{z}$

Notice that $\mathbf{z}$ could also be fixed by hand. However, the main challenge of hand-design is to correctly choose the offsets. For example, a randomly-selected $\mathbf{z}$ may not incorporate frequency dependency. In addition, searching over all M^D possible configurations would be cumbersome. For these reasons, Bresenham, D-Rational and U-Rational strategies were elaborated, as seen in Fig. 2. By defining a curve that depends on frequency, the position offsets can be found using the closest points to the curve.

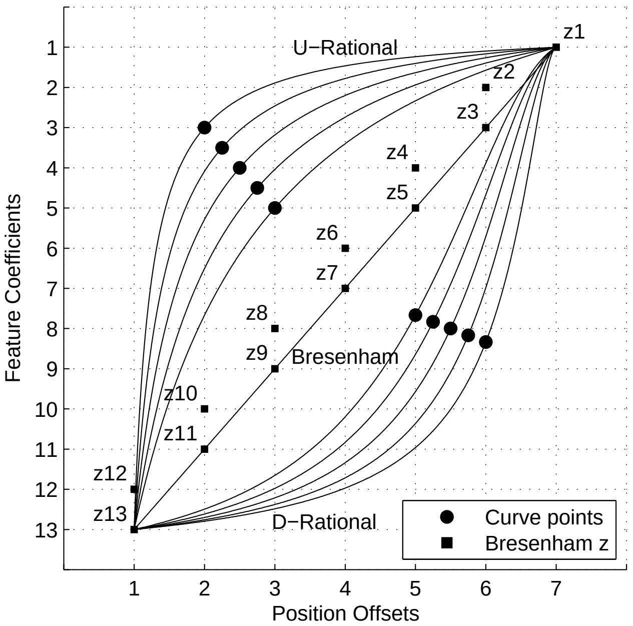


Fig. 2 Example of an experimental setup that uses a maximum position offset $k = 7$. For Bresenham, the offsets $\mathbf{z} = [7, 6, 6, 5, 5, 4, 4, 3, 3, 2, 2, 1, 1]$.

For *Bresenham*, a maximum position offset k is fixed. Then, Bresenham's line algorithm [Bresenham, 1965] is applied on the line that starts at coefficient 13 and time offset 1, and ends at coefficient 1 and time offset k. The points found by Bresenham's algorithm correspond to $\mathbf{z}$.

For the **-Rational* strategies, a maximum position offset k and a curve point c are fixed. The offsets $\mathbf{z}$ can then be inferred from the rational function that starts at coefficient 13 and time offset 1, and ends at coefficient 1 and offset k, with the additional constraint that it passes through the curve point c. When c is above the line, we refer to this strategy as *U-Rational*, otherwise as *D-Rational*. Depending on the degrees of the polynomials used in the rational functions, the curves will have distinct shapes and reflect different types of frequency dependency.

3.4 Decorrelation

The most common inference model that is used in ASR is the hidden Markov model (HMM), with a mixture of Gaussian distributions for the observation density (GMM-HMM). It is usually assumed that the multivariate Gaussian distributions have a diagonal covariance

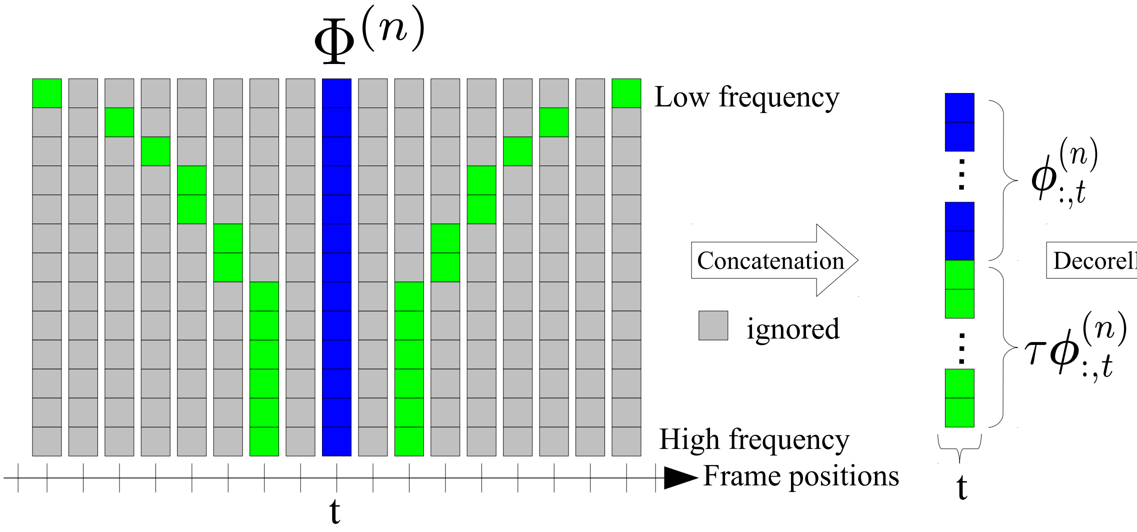


Fig. 3 Pipeline of processing for TFS features. After computing the frame position offsets $\mathbf{z}$ using Eq. 8, the features are concatenated and decorrelated (using DCT-II, ICA or whitening).

[Gales and Young, 2008]. This hypothesis is required so that the model can scale to large-vocabulary continuous speech recognition (LVCSR). The consequence of this assumption is that the feature frames should be independent, which is clearly not the case for TFS features.

Therefore, the feature frames are decorrelated after the concatenation in order to accommodate them to the independence hypothesis, as seen in Fig 3 . Three methods of decorrelation have been used in our experimentations: discrete cosine transform (DCT), whitening (W) and independent components analysis (ICA) [Hyvärinen et al., 2004]. Specifically, the type 2 DCT was chosen (DCT-II) and a *logcosh* G function was used in the approximation of the neg-entropy for ICA.

4 Experimental Results

4.1 Experimental Setup

The database that we used for our experiments is Aurora 2 [Pearce et al., 2000] which contains a vocabulary of 11 spoken digits (*zero* to *nine* with *oh*). The digits are connected, thus they can be spoken in any order and in any amount (up to 7) with possible pauses between them. The training set contains 8,440 utterances, both test set A and B have 28,028 and test C has 14,014 utterances. The utterances are noisy and the signal-to-noise ratio (SNR) varies from -5 dB, 0 dB, . . . , 20 dB, Inf dB (clean). Different kinds of noise are present such as train, airport, car, restaurant, etc. On average, an utterance lasts approximately 2 seconds.

Using the HTK [Young et al., 2006] framework provided with the Aurora 2 database, we performed two experiments. In the first one, eighteen-states whole-word HMMs were trained with a three-components GMM as the state emission density. There was a total of 11 HMMs (one per class). In the second one, the whole-word HMMs were replaced with five-states phoneme HMMs. In other words, using the CMU pronouncing dictionary, each digit was mapped to its ARPAbet interpretation. There was a total of 19 HMMs (one per phoneme).

In our experimentations, we compared TFS features to first (-D) and second (-A) order delta features on MFCC, PLPCC and LPCC. For all these features, 13 coefficients, including the energy (-E), excluding the 0th coefficient, were extracted to be used as observations. The performance of each method was averaged over all test sets for each noise level separately.

We tested multiple configurations of $\mathbf{z}$ and decorrelation methods. First, we evaluated the proposed learning framework described in Section 3.2 with $V_{thresh} = 1$ using DCT decorrelation. In addition, when the configuration $\mathbf{z}$ is fixed for all utterances, we refer to it with suffix -T (MFCC-T, PLPCC-T and LPCC-T). When it is computed for each utterance independently (therefore variable), we refer to it with suffix -δT (MFCC-δT, PLPCC-δT and LPCC-δT).

We also tested the three different hand-designed strategies of Section 3.3. The reader can refer to Fig. 2 that provides additional information relative to the different approaches. For Bresenham, 20 maximum offsets k were tested, $k \in \{1, \ldots, 20\}$, and the one achieving the best result is reported. For U-Rational, the curve is modeled with a rational function using a polynomial of degree 1 over a polynomial of degree 1. For D-Rational, a poly-

Features	Frame position offsets **z**	SNR (dB)							Accuracy (%)	
		Inf	20	15	10	5	0	-5	Avg	R.I.
MFCC-E-D-A	-	**98.54**	97.14	96.02	93.27	84.86	57.47	23.35	78.66	-
MFCC-E-T	$[8,6,5,4,4,3,3,2,2,2,2,2,2]$	97.64	97.46	96.68	94.39	88.03	**71.31**	38.93	**83.49**	**22.63**
MFCC-E-δT	-	97.20	96.98	96.29	93.78	87.27	69.20	35.90	82.37	17.39
Bresenham DCT	$[5,5,4,4,4,3,3,3,2,2,2,1,1]$	97.53	97.48	96.58	94.26	87.99	71.22	39.08	83.45	22.45
U-Rational DCT	$[7,5,4,4,3,3,2,2,2,1,1,1,1]$	97.53	97.40	96.51	94.30	87.92	70.87	38.59	83.30	21.74
D-Rational DCT	$[8,7,7,7,6,6,5,5,5,4,3,2,1]$	97.06	96.91	95.92	93.63	87.34	70.63	39.70	83.03	20.48
Bresenham ICA	$[6,6,5,5,4,4,3,3,3,2,2,1,1]$	98.39	**98.36**	**97.64**	**95.59**	**89.00**	69.27	32.22	82.93	20.01
U-Rational ICA	$[8,6,5,4,3,3,2,2,2,2,1,1,1]$	98.45	98.34	97.56	95.47	88.94	69.12	32.47	82.91	19.92
D-Rational ICA	$[7,7,7,6,6,6,6,6,6,5,5,4,1]$	98.22	98.01	97.10	95.03	88.06	68.81	31.98	82.46	17.81
Bresenham W	$[5,5,4,4,4,3,3,3,2,2,2,1,1]$	96.96	96.77	95.56	92.74	86.42	69.54	39.92	82.56	18.28
U-Rational W	$[9,5,3,3,2,2,2,1,1,1,1,1,1]$	96.79	97.19	96.26	93.85	87.39	70.30	39.14	82.99	20.29
D-Rational W	$[7,7,6,6,6,6,5,5,5,4,4,3,1]$	96.23	95.99	94.85	91.93	85.37	69.65	**41.22**	82.18	16.49

Table 1 Word accuracy (%) of different **z** using 13 MFCC-E features on the Aurora 2 database for whole-word HMMs. The results are averaged according to the noise level. The reference model is MFCC-E-D-A.

nomial of degree 1 over a polynomial of degree 2 was used. 4 maximum offsets k, $k \in \{7, 8, 9, 10\}$, and 5 curve points c were tested and the one achieving the best result is reported. For each strategy, the coefficients from each feature frame are concatenated, transformed using one of the three decorrelation methods and each utterance is standardized independently.

4.2 Experimental Results

The performances in word accuracy of the best configurations of **z** are reported in Tables 1, 2 and 3 for whole-word HMMs and Tables 4, 5 and 6 for phoneme HMMs. In each table, the 7 noise levels from the Aurora 2 database are ordered from clean signals (SNR Inf) to highly noisy signals (SNR -5). The average over all noise levels is reported on the right. The last column consists of the relative improvement of the method over the reference model.

Based on these results, our approach for learning the time offsets **z** achieved the best average relative improvement of 20.79% for whole-word HMM and 32.10% for phoneme HMM. Also, it can be observed that the TFS features increased the accuracy on almost all noisy tasks (Table 3 shows that Bresenham DCT is better on average). However, the TFS features did not improve the performances of whole-word HMMs on clean signals. Nonetheless, these results support our initial intuition that using a pure derivative approach leads to inferior performances.

The variation of the word accuracy of MFCC-E-T, PLPCC-E-T and LPCC-E-T, with respect to V_{thresh}, is shown in Fig. 4 for whole-word HMMs. The performance of the method is reported for the 7 noise levels of the database. The crosses indicate the best result the method achieved for each noise level. This figure demonstrates the behavior of the performance of our approach with respect to the parametrization of **z**. We only reported the variation for -T features, but the other methods displayed similar behavior.

Features	Frame position offsets **z**	SNR (dB)							Accuracy (%)	
		Inf	20	15	10	5	0	-5	Avg	R.I.
PLPCC-E-D-A	-	**98.65**	97.56	96.48	93.85	85.93	59.83	25.36	79.66	-
PLP-E-T	$[8,6,5,4,4,3,3,3,3,3,2,2,2]$	97.40	97.36	96.60	94.52	88.43	**71.98**	40.23	**83.79**	**20.30**
PLPCC-E-δT	-	97.33	97.18	96.55	94.24	87.96	70.28	36.99	82.93	16.08
Bresenham DCT	$[6,6,5,5,4,4,3,3,3,2,2,1,1]$	97.63	97.44	96.67	94.49	88.32	71.84	39.14	83.65	19.62
U-Rational DCT	$[8,6,5,4,3,3,2,2,2,2,1,1,1]$	97.60	97.47	96.72	94.60	88.30	71.16	38.67	83.50	18.88
D-Rational DCT	$[7,7,6,6,6,6,5,5,5,4,4,3,1]$	97.18	97.05	96.19	94.00	88.02	71.81	**40.45**	83.53	19.03
Bresenham ICA	$[7,6,6,5,5,4,4,3,3,2,2,1,1]$	98.46	98.38	97.46	95.39	**88.70**	69.29	33.92	83.08	16.81
U-Rational ICA	$[8,6,4,3,3,2,2,2,2,1,1,1,1]$	98.46	**98.44**	**97.65**	**95.63**	88.53	68.52	33.01	82.89	15.88
D-Rational ICA	$[7,7,7,6,6,6,6,6,6,5,5,4,1]$	98.22	98.04	97.06	95.06	88.22	69.46	34.19	82.89	15.88
Bresenham W	$[4,4,3,3,3,3,2,2,2,2,1,1,1]$	97.32	97.46	96.42	93.62	85.83	65.49	32.06	81.17	7.42
U-Rational W	$[7,4,3,3,2,2,2,1,1,1,1,1,1]$	97.01	97.41	96.49	93.74	85.86	64.76	30.48	80.82	5.70
D-Rational W	$[7,7,6,6,6,6,5,5,5,4,4,3,1]$	96.71	96.76	95.50	92.44	84.32	64.44	32.97	80.45	3.88

Table 2 Word accuracy (%) of different **z** using 13 PLPCC-E features on the Aurora 2 database for whole-word HMMs. The results are averaged according to the noise level. The reference model is PLPCC-E-D-A.

Features	Frame position offsets **z**	SNR (dB)							Accuracy (%)	
		Inf	20	15	10	5	0	-5	Avg	R.I.
LPCC-E-D-A	-	**98.30**	96.82	95.59	92.28	81.87	54.52	22.96	77.48	-
LPCC-E-T	$[8, 6, 5, 5, 4, 4, 3, 3, 2, 2, 2, 2, 2]$	96.74	96.90	95.90	93.30	85.91	67.86	36.39	81.86	19.45
LPCC-E-δT	-	96.31	96.60	95.62	92.72	85.16	65.88	33.07	80.77	14.61
Bresenham DCT	$[7, 6, 6, 5, 5, 4, 4, 3, 3, 2, 2, 1, 1]$	96.71	96.98	96.03	93.26	86.06	**68.14**	36.53	**81.96**	**19.89**
U-Rational DCT	$[7, 5, 4, 4, 3, 3, 2, 2, 2, 1, 1, 1, 1]$	96.53	96.85	95.86	93.02	85.83	67.93	36.32	81.76	19.01
D-Rational DCT	$[9, 8, 8, 7, 7, 6, 5, 5, 4, 4, 3, 2, 1]$	96.30	96.68	95.51	92.70	85.24	67.83	**36.75**	81.57	18.16
Bresenham ICA	$[5, 5, 4, 4, 4, 3, 3, 3, 2, 2, 2, 1, 1]$	98.03	98.09	**97.14**	**94.81**	**87.16**	66.88	30.39	81.78	19.09
U-Rational ICA	$[8, 6, 5, 4, 3, 3, 2, 2, 2, 2, 1, 1, 1]$	97.82	**97.87**	96.93	94.63	87.01	66.62	30.11	81.57	18.16
D-Rational ICA	$[10, 9, 8, 7, 7, 6, 5, 4, 4, 3, 2, 2, 1]$	97.80	97.32	96.24	93.54	85.92	67.53	34.02	81.77	19.05
Bresenham W	$[4, 4, 3, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1]$	96.71	96.66	95.46	92.26	84.70	66.00	34.93	80.96	15.45
U-Rational W	$[7, 5, 4, 4, 3, 3, 2, 2, 2, 1, 1, 1, 1]$	96.52	96.73	95.52	92.34	84.48	65.56	33.86	80.72	14.39
D-Rational W	$[8, 7, 7, 7, 6, 6, 5, 5, 5, 4, 3, 2, 1]$	95.97	95.92	94.49	91.30	83.78	65.75	36.21	80.49	13.37

Table 3 Word accuracy (%) of different **z** using 13 LPCC-E features on the Aurora 2 database for whole-word HMMs. The results are averaged according to the noise level. The reference model is LPCC-E-D-A.

5 Discussion

Based on the results in Tables 1–6, DCT is a better decorrelation method than whitening and ICA. In most cases, the average word accuracy is higher when DCT is used to decorrelate the TFS features. The only exceptions were D-Rational in Table 3, Bresenham in Table 4 and both Bresenham and U-Rational in Table 6, where ICA based features achieved higher average word accuracy. ICA only achieved the best performances when the utterances are slightly noisy (SNR greater than 5 dB). The susceptibility to noise is one of the method's limitations. Indeed, the standard definition of the approach does not define a noise term. This requires the use of whitening prior to the decomposition when noisy data are used. However, the results show that ICA can not extract useful independent components when the utterances have an SNR lower than 10 dB.

One limitation of the proposed TFS method is that it does not outperform delta features for clean utterances when using whole-word HMMs. This can be seen in Tables 1–3 where, for clean utterances (SNR Inf), the delta features achieve the best accuracy. These results are consistent with the intuition given in Section 1 that the derivative of a noisy signal amplifies the noise. In the case of clean utterances, the derivative is a better approach for extracting the dynamics of the signal.

However, when using phoneme HMMs, our results suggest that the TFS method improves word accuracy even in the absence of noise. This can be seen in Tables 4–6 where -T is always better than -D-A. These non intuitive results could be related to using phonemes instead of whole words to model speech. Indeed, TFS appears to simulate triphone modeling, where an HMM is defined for every phoneme triplet. In other words, TFS incorporates information about adjacent phonemes when concatenating distanced coefficients. This was not the case with whole-word HMMs because the state occupancy of a phoneme HMM is usually much shorter.

Moreover, it can be seen that computing the position offsets **z** for each utterance separately (-δT) yields much lower word accuracy. The reason explaining these bad performances may be related to the use of GMM as the acoustic model. By changing the positions at which

Features	Frame position offsets **z**	SNR (dB)							Accuracy (%)	
		Inf	20	15	10	5	0	-5	Avg	R.I.
MFCC-E-D-A	-	89.89	87.24	84.41	78.87	63.78	29.86	-5.82	61.17	-
MFCC-E-T	$[8, 6, 5, 4, 4, 3, 3, 2, 2, 2, 2, 2, 2]$	93.02	94.15	92.65	88.84	79.22	56.42	19.58	**74.84**	**35.20**
MFCC-E-δT	-	92.44	93.6	91.7	87.55	76.46	47.3	-0.88	69.74	22.07
Bresenham DCT	$[8, 7, 7, 6, 6, 5, 4, 4, 3, 3, 2, 2, 1]$	93.00	94.05	92.54	88.58	79.19	56.51	19.07	74.70	34.84
U-Rational DCT	$[8, 6, 5, 4, 3, 3, 2, 2, 2, 2, 1, 1, 1]$	93.03	94.07	92.6	88.82	78.88	56.07	19.66	74.73	34.92
D-Rational DCT	$[9, 8, 8, 7, 7, 6, 5, 5, 4, 4, 3, 2, 1]$	93.01	93.58	91.86	88.01	78.74	**57.13**	21.49	74.83	35.17
Bresenham ICA	$[8, 7, 7, 6, 6, 5, 4, 4, 3, 3, 2, 2, 1]$	94.45	95.02	93.47	89.36	78.46	52.83	20.02	74.80	35.10
U-Rational ICA	$[8, 6, 5, 4, 3, 3, 2, 2, 2, 2, 1, 1, 1]$	92.26	94.21	92.95	89.46	77.66	47.29	7.27	71.59	26.83
D-Rational ICA	$[7, 7, 7, 6, 6, 6, 6, 6, 6, 5, 5, 4, 1]$	**94.7**	**95.86**	**94.35**	**90.33**	**79.3**	50.59	12.32	73.92	32.84
Bresenham W	$[6, 6, 5, 5, 4, 4, 3, 3, 3, 2, 2, 1, 1]$	89.75	90.36	87.98	83.26	73.23	51.18	**22.08**	71.12	25.62
U-Rational W	$[8, 5, 4, 3, 3, 2, 2, 2, 1, 1, 1, 1, 1]$	89.57	90.54	88.21	83.74	73.55	49.18	17.76	70.36	23.67
D-Rational W	$[10, 9, 8, 7, 7, 6, 5, 4, 4, 3, 2, 2, 1]$	88.33	89.18	86.78	82.18	72.11	50.88	21.09	70.08	22.95

Table 4 Word accuracy (%) of different **z** using 13 MFCC-E features on the Aurora 2 database for phoneme HMMs. The results are averaged according to the noise level. The reference model is MFCC-E-D-A.

Features	Frame position offsets **z**	SNR (dB)							Accuracy (%)	
		Inf	20	15	10	5	0	-5	Avg	R.I.
PLPCC-E-D-A	-	88.99	87.92	84.78	78.97	64.18	32.96	-0.67	62.45	-
PLPCC-E-T	$[8,6,5,4,4,3,3,3,3,3,2,2,2]$	92.98	94.29	92.89	89.38	79.76	56.86	18.79	**74.99**	**33.40**
PLPCC-E-δT	-	92.27	93.77	92.2	88.17	77.59	48.83	-1.96	70.12	23.05
Bresenham DCT	$[7,6,6,5,5,4,4,3,3,2,2,1,1]$	92.6	94.52	93.13	89.24	80.08	57.05	18.09	74.95	33.29
U-Rational DCT	$[8,6,4,3,3,2,2,2,2,1,1,1,1]$	91.91	93.89	92.33	88.57	79.11	56.24	**19.88**	74.56	32.25
D-Rational DCT	$[8,7,7,7,6,6,5,5,5,4,3,2,1]$	92.08	94.1	92.63	88.98	**80.14**	**57.56**	19.24	74.95	33.29
Bresenham ICA	$[9,8,8,7,6,6,5,4,4,3,2,2,1]$	93.8	94.3	92.71	89.02	78.26	52.6	19.19	74.26	31.45
U-Rational ICA	$[8,6,5,4,3,3,2,2,2,2,1,1,1]$	93.06	94.41	92.89	88.98	77.26	49.9	14.39	72.98	28.04
D-Rational ICA	$[7,7,6,6,6,6,6,5,5,5,4,3,1]$	**94.57**	**95.85**	**94.38**	**90.39**	78.97	51.2	13.55	74.12	31.08
Bresenham W	$[5,5,4,4,4,3,3,3,2,2,2,1,1]$	90.32	91.46	89.66	84.46	70.78	41.01	0.45	66.88	11.80
U-Rational W	$[9,5,3,3,2,2,2,1,1,1,1,1,1]$	88.41	90.63	89.09	84.36	71.24	41.32	-1.32	66.25	10.12
D-Rational W	$[7,7,6,6,6,6,5,5,5,4,4,3,1]$	90.14	90.52	88.92	83.75	69.64	40.9	-0.37	66.21	10.01

Table 5 Word accuracy (%) of different **z** using 13 PLPCC-E features on the Aurora 2 database for phoneme HMMs. The results are averaged according to the noise level. The reference model is PLPCC-E-D-A.

adjacent coefficients are selected, the isotropic Gaussian densities of the mixture no longer well represent the speech. The reason is that the feature space defined by all feature frames of all utterances does not accommodate the independence hypothesis (defined in Section 3.4) even though decorrelation is performed. This is not the case for the original formulation (-T).

The first interesting result that is worth noticing from Fig. 4 is the convexity of the plots. The up and down hill-shaped curves are experimental supports for the idea of informative coefficients that was elaborated in Sections 1 and 3. If the concatenated coefficients are taken too close, or too far, from each other, the amount of unrelated information added to the frame will be greater than the amount of related information. This phenomenon can be observed in Fig. 4 where the performance increases as V_{thresh} increases, up to a certain point where it starts to decrease.

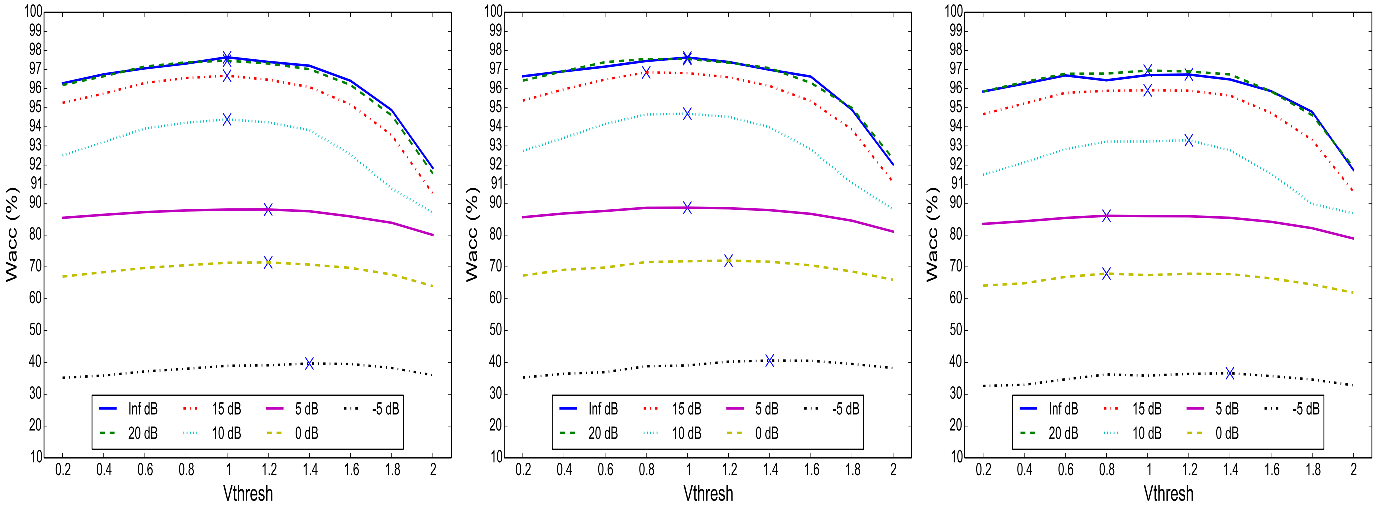
The second result that is worth mentioning is the behavior of the maximum accuracy with respect to the noise level. As the noise increases, the maximum word accuracy tends to occur for **z** that has greater time offsets. For example, the best word accuracy for Fig. 4 (a) appears at $V_{thresh} = 1$ for the least noisy task and at $V_{thresh} = 1.4$ for the noisiest one. It appears that our approach acts like a noise reduction method by smoothing the signal (like Gaussian blur). Smoothing a highly noisy signal requires gathering information at a farther distance. Our approach behaves similarly by selecting coefficients that are farther apart.

It is also worth mentioning that this behavior is less clearly identifiable for LPCC, as can be seen in Fig. 4(c). While the reason explaining this phenomenon is not evident, the word accuracy of tasks with an SNR lower than 10 dB does not vary significantly with respect to V_{thresh}. This may suggest that the TFS approach extracts less relevant dynamical information from LPCC. It could also be related to the fact that LPCC almost always performs worse than PLPCC and MFCC, as shown in Tables 1–6.

To sum up, our results suggest that the concatenation of adjacent coefficients based on frequency helps improve the accuracy, especially for noisy utterances. It has been observed, based on our experimentations,

Features	Frame position offsets **z**	SNR (dB)							Accuracy (%)	
		Inf	20	15	10	5	0	-5	Avg	R.I.
LPCC-E-D-A	-	88.13	86.04	83.11	76.64	60.82	29.65	0.02	60.63	-
LPCC-E-T	$[8,6,5,5,4,4,3,3,2,2,2,2,2]$	91.09	93.26	91.24	86.98	76.45	50.90	10.79	**71.53**	**27.69**
LPCC-E-δT	-	90.11	92.23	90.06	84.86	71.79	37.92	-25.48	63.07	6.20
Bresenham DCT	$[5,5,4,4,4,3,3,3,2,2,2,1,1]$	91.44	93.38	91.47	86.74	75.37	49.04	8.00	70.78	25.78
U-Rational DCT	$[8,6,5,4,3,3,2,2,2,2,1,1,1]$	91.18	92.94	90.94	86.42	75.31	49.69	8.67	70.73	25.65
D-Rational DCT	$[10,9,8,7,7,6,5,4,4,3,2,2,1]$	91.06	92.81	90.85	86.27	75.36	50.97	11.89	71.31	27.13
Bresenham ICA	$[5,5,4,4,4,3,3,3,2,2,2,1,1]$	92.47	91.99	89.76	84.8	72.69	48.21	15.87	70.82	25.88
U-Rational ICA	$[7,5,4,3,3,2,2,2,2,1,1,1,1]$	89.16	92.53	91.16	86.95	74.56	47.91	18.09	71.48	27.56
D-Rational ICA	$[10,10,10,10,10,9,9,9,9,8,7,6,1]$	92.09	92.3	89.95	84.82	72.05	47.01	14.42	70.37	24.74
Bresenham W	$[6,6,5,5,4,4,3,3,3,2,2,1,1]$	88.49	89.81	87.31	81.6	68.33	42.66	7.76	66.56	15.06
U-Rational W	$[7,5,4,3,2,2,2,2,1,1,1,1,1]$	88.35	90.25	88.07	82.8	68.85	40.94	3.8	66.15	14.02
D-Rational W	$[7,7,6,6,6,6,5,5,5,4,4,3,1]$	88.84	89.65	87.13	81.18	68.00	42.11	6.70	66.23	14.22

Table 6 Word accuracy (%) of different **z** using 13 LPCC-E features on the Aurora 2 database for phoneme HMMs. The results are averaged according to the noise level. The reference model is LPCC-E-D-A.



(a) Using MFCC features.

(b) Using PLPCC features.

(c) Using LPCC features.

Fig. 4 Variation of the performance of -T features for whole-word HMMs on the Aurora 2 database. The crosses indicate the maximum word accuracy for each noise level.

that our TFS features achieved a better over all performance than the delta features. However, it is worth mentioning that, for clean utterances with whole-word HMMs, our method did not outperform the reference features. Nonetheless, TFS appears to be a good choice for dynamical features since it performed the best overall, can be learned rapidly from the data and is based on a single specified parameter V_{thresh}.

6 Conclusion

A novel way of improving the dynamics of static speech features was proposed. The issue that was addressed was the susceptibility to noise of derivative operations in the modeling of the dynamics of speech signals. The proposed Temporal Feature Selection (TFS) features have shown to improve the robustness of the state of the art delta features in various types of noise. The experimentations have shown that the three most standard features, MFCC, PLPCC and LPCC, combined with the best TFS features achieved an average relative improvement of 20.79% and 32.10% in accuracy for whole-word and phoneme HMMs on the Aurora 2 database.

For further study, we plan to evaluate our approach using triphone HMMs. The results would indicate if TFS can incorporate information about adjacent phonemes when concatenating distanced feature frames. Additionally, we intend to use a Deep Neural Network as the acoustic model in order to evaluate the influence of TFS on splicing. We believe that a smaller context window could be used and thus reduce the input dimensionality. Finally, more experiments could be performed on large vocabulary continuous speech recognition tasks.

References

[Bahl et al., 1994] Bahl, L., De Souza, P., Gopalakrishnan, P., Nahamoo, D., and Picheny, M. (1994). Robust methods for using context-dependent features and models in a continuous speech recognizer. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 1, pages I–533. IEEE.

[Bresenham, 1965] Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Syst. J.*, 4(1):25–30.

[Fukunaga, 1990] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition (2nd Ed.)*. Academic Press Professional, Inc., San Diego, CA, USA.

[Furui, 1986] Furui, S. (1986). Speaker-independent isolated word recognition based on emphasized spectral dynamics. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, volume 11, pages 1991–1994. IEEE.

[Gales and Young, 2008] Gales, M. and Young, S. (2008). The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304.

[Gales, 1998] Gales, M. J. (1998). Maximum likelihood linear transformations for hmm-based speech recognition. *Computer speech & language*, 12(2):75–98.

[Gales, 1999] Gales, M. J. (1999). Semi-tied covariance matrices for hidden markov models. *Speech and Audio Processing, IEEE Transactions on*, 7(3):272–281.

[Gopinath, 1998] Gopinath, R. A. (1998). Maximum likelihood modeling with gaussian distributions for classification. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 661–664. IEEE.

[Hossan et al., 2010] Hossan, M. A., Memon, S., and Gregory, M. A. (2010). A novel approach for MFCC feature extraction. In *Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*, pages 1–5. IEEE.

[Hyvärinen et al., 2004] Hyvärinen, A., Karhunen, J., and Oja, E. (2004). *Independent component analysis*, volume 46. John Wiley & Sons.

[Jolliffe, 1986] Jolliffe, I. (1986). Principal component analysis. *Springer Series in Statistics, Berlin: Springer, 1986*, 1.

[Kumar et al., 2011] Kumar, K., Kim, C., and Stern, R. M. (2011). Delta-spectral cepstral coefficients for robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 4784–4787. IEEE.

[Kumar and Andreou, 1998] Kumar, N. and Andreou, A. G. (1998). Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech communication*, 26(4):283–297.

[Leggetter and Woodland, 1995] Leggetter, C. J. and Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2):171–185.

[Lockwood and Boudy, 1992] Lockwood, P. and Boudy, J. (1992). Experiments with a nonlinear spectral subtractor (NSS), Hidden Markov models and the projection, for robust speech recognition in cars. *Speech Communication*, 11(23):215 – 228.

[Oppenheim et al., 1999] Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (1999). *Discrete-time Signal Processing (2nd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Pearce et al., 2000] Pearce, D., günter Hirsch, H., and Gmbh, E. E. D. (2000). The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ISCA ITRW ASR2000*, pages 29–32.

[Rath et al., 2013] Rath, S. P., Povey, D., and Veselỳ, K. (2013). Improved feature processing for deep neural networks. In *Proc. Interspeech*.

[Saon et al., 2000] Saon, G., Padmanabhan, M., Gopinath, R., and Chen, S. (2000). Maximum likelihood discriminant feature spaces. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 2, pages II1129–II1132. IEEE.

[Shrawankar and Thakare, 2013] Shrawankar, U. and Thakare, V. M. (2013). Techniques for feature extraction in speech recognition system: A comparative study. *arXiv:1305.1145*.

[Trottier et al., 2014] Trottier, L., Chaib-draa, B., and Giguère, P. (2014). Effects of frequency-based inter-frame dependencies on automatic speech recognition. In *Canadian Conference on AI*, pages 357–362.

[Weng et al., 2010] Weng, Z., Li, L., and Guo, D. (2010). Speaker recognition using weighted dynamic MFCC based on GMM. In *Anti-Counterfeiting Security and Identification in Communication (ASID), 2010 International Conference on*, pages 285–288. IEEE.

[Young et al., 2006] Young, S. J., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. C. (2006). *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK.

[Yu et al., 2013] Yu, D., Seltzer, M. L., Li, J., Huang, J.-T., and Seide, F. (2013). Feature learning in deep neural networks-studies on speech recognition tasks. *arXiv:1301.3605*.

[Zheng et al., 2001] Zheng, F., Zhang, G., and Song, Z. (2001). Comparison of different implementations of MFCC. *Journal of Computer Science and Technology*, 16(6):582–589.