Delta Vector Taylor Series Environment Compensation for Speaker Recognition

Brian Eberman and Pedro J. Moreno
email: bse@crl.dec.com, pjm@crl.dec.com

Digital Equipment Corporation
Cambridge Research Laboratory

ABSTRACT

The performance of speaker recognition algorithms drops significantly when testing and training acoustic environments differ. This decrease is caused by the statistical mismatch between the statistics representing the speaker and the testing acoustic data. This paper reports our preliminary results on the application of a novel environmental compensation algorithm to the problem of speaker recognition and identification. This new technique, called the Delta Vector Taylor Series (DVTS) approach, improves performance at signal-to-noise ratios below 20dB. The algorithm imposes a model of how the environment modifies speaker statistics and uses Expectation-Maximization (EM) to solve a joint maximum likelihood formulation for the speaker recognition problem over both the speakers and the environment. We report experimental results on a subset of the TIMIT and NTIMIT database.

1. Introduction

As speaker recognition technology is deployed in richer environments robustness to the effects of the environment becomes increasingly important. In many situations the statistical differences between the training corpus used to create speaker models and the testing utterances used to verify/identify speakers can be quite dramatic. It is therefore important to develop algorithms able to cope with this statistical mismatch.

Over the past serveral years there has been considerable work on speech recognition robustness. In speaker recognition there has been work on robust model training, and in robust scoring techniques [4], but relatively little work on remapping of the speech statistics to reflect changes in the channel or overall background noise.

Zhang and Mammone [10] describe one approach for environmental compensation. In their technique, the effect of the channel and noise on the cepstra is modeled using an affine transform. The transformation is similar to the MLLR transformation used by Leggetter and Woodland [7] for robust speech recognition. In their approach, Zhang and Mammone derive analytically the appropriate affine transform for additive, Gaussian, white noise and a linear channel distortion. In their experiments, the desired affine transformation is then determined from the known channel and noise conditions and applied to a Vector Codebook model of each speaker. The speaker which had the minimum accumulated distance between their transformed codebook and the measured speech is then selected as the identified speaker.

This paper presents a different approach to the problem of environmental robustness in speaker reognition. Rather than using an affine transformation for modelling the effect of the environment on speech statistics (a codebook) we use an analytical model of how additive noise and convolutional noise affect speech statistics. We will show how the vector codebook approach can be made into a blind matching approach by: 1) using a non-linear model for the effect of the environment on features computed using a FFT, and 2) using the EM algorithm to do speaker dependent maximization of the environmental parameters. Gish's ML [4] statistic is then computed from the transformed codebook and used for identifying the speaker. However, any final speaker scoring algorithm could be used including Bimbot's AHS metric [2] or a speaker mixture model [9]. The technique improves performance even though it broadens each speakers model by independently adaptating them to the target speech.

2. Background

The DVTS technique is based on an extension of the Vector Taylor Series (VTS) [8] approach to robust speech recognition. This technique models the effect of the environment on speech feature vectors by introducing an environment function $\mathbf{z} = \mathbf{x} + \mathbf{f}(\mathbf{x}, \mathbf{n}, \mathbf{q})$, where $\mathbf{z}$ represents the measured noisy speech feature vector, $\mathbf{x}$ represents a clean speech feature vector, $\mathbf{f}$ is the environmental function, and $(\mathbf{n}, \mathbf{q})$ represent the environmental parameters. In our case, we assume a simple model of the environment as additive, Gaussian, noise and linear filtering. This model was original proposed by Acero [1] and later used by Gales [3] and others.

Given the statistics of the speaker $p(\mathbf{x}|s)$, based on clean speech vectors, the above equation can be used to compute the statistics of the speaker under distorted conditions $p(\mathbf{z}|s)$. However, if $p(\mathbf{x}|s)$ is modeled as a Gaussian mixture, there is no known analytical technique for computing $p(\mathbf{z}|s)$.

The VTS approach solved this problem by approximating $\mathbf{f}(\mathbf{x}, \mathbf{n}, \mathbf{q})$ by a vector Taylor series about the mean of each mixture component. The linearization of $\mathbf{f}$ makes

the computation of $p(\mathbf{z}|s)$ a simple, mixture component dependent linear transformation. However, the accuracy of VTS is constrained by the magnitude of the covariance of the mixtures. The covariance controls the size of the second order effects in the Taylor series, and the larger the covariance the less accurate the transformation.

DVTS takes the VTS idea to its logical extension by modeling the statistics $p(\mathbf{x}|s)$ for a speaker s by the training speech available from that speaker, or via a large vector-codebook trained on the speaker's speech. The underlying speech statistics can be thought of as a collection of weighted Dirac deltas, $p(\mathbf{x}|s) = \sum_{k=1}^{K} P[k,s]\ \delta(\mathbf{x} - \mathbf{v}(k,s))$. The delta codebook represents an idealization of the speaker's speech that is useful in simplifying the math.

The process of measuring the speech will always broaden the delta's distribution transforming the measured speech distribution $p(\mathbf{z})$ into a mixture of Gaussians. In fact it can be shown, under standard models for speech, that the process of measuring the spectrum of the speech using a fixed window and FFT has a minimum covariance that depends only upon the mel-smoothing and is independent of the window size or the frequency content of the signal.

Thus, conceptually our approach is to model the source speech as a collection of very narrow distributions (the Dirac deltas) the covariance of which can be neglected in the environmental transformation, but the effect of the environmental/measurement transformation will be to smooth the source distribution producing a Gaussian mixture model with a single covariance shared across all mixture components.

This is probably not a good model for speech recognition or for speaker modeling, because there the problem is smoothing the sample statistics to obtain a good approximation to $p(\mathbf{x}|s)$. However it is an excellent model for environmental estimation, because in this case we are only trying to find a few overall environmental parameters and the sampling process will smooth the effects of the delta modeling. The simplicity of the model reduces the mathematical manipulations required for implementation and opens up the possibility for a richer environmental model, *i.e.*, environmental models different from the additive noise and linear filtering by an unknown channel.

3. Algorithm

In our system the mel-frequency spectral coefficients (MFSC) are computed using a 410 point Hamming window and a 512 point FFT. The resulting power spectra are reduced to 41 mel-frequency power terms, then the log is taken to get the MFSC components. Based on this signal processing, f takes the form

$$\mathbf{z} = \log(\exp(\mathbf{x} + \mathbf{q}) + \exp(\mathbf{n})) \tag{1}$$

where $\mathbf{x}$ and $\mathbf{z}$ are the MFSC vectors, $\mathbf{q}$ is the MFSC channel, $\mathbf{n}$ is the power of the noise in the MFSC domain. The functions are applied component-wise.

Performing a first order Taylor expansion of equation 1 about each training vector, or codebook vector, $\mathbf{v}(k,s)$

$$\begin{aligned} \mathbf{E}[\mathbf{z}|k,s] &= \mathbf{v}(k,s) + \mathbf{f}(\mathbf{v}(k,s),\mathbf{n},\mathbf{q}) && (2)\\ \Sigma[\mathbf{z}|k,s] &= \mathbf{B}^T \Sigma_n \mathbf{B} && (3)\\ \mathbf{B} &= \mathrm{diag}(1/(\mathbf{b}(k,s)+1)) && (4) \end{aligned}$$

for the expected value and covariance of $\mathbf{z}$ about each vector, and $\mathbf{b}(k,s) = \exp(\mathbf{q}+\mathbf{v}(k,s)-\mathbf{E}[\mathbf{n}])$ is the effective signal-to-noise ratio at a training vector [1]. The covariance of the measurement about a given vector depends upon the local signal-to-noise ratio, however, we will assume that this covariance is constant. The environment, given a speaker, is then represented by $\theta = (\mathbf{q}, \mathbf{n}, \Sigma_z)$.

The log-likelihood of an ensemble of N measurements $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ given a putative speaker s, is defined as $l(\mathcal{Z}|s,\theta) = \sum_{t=1}^{N} \log(p(\mathbf{z}_t|s,\theta))$. The EM algorithm maximizes this by iteratively solving an auxiliar function $\mathcal{Q}$ defined as

$$\mathcal{Q}(\theta,\theta'|s) = \sum_{t=1}^{N}\sum_{k=1}^{K} P[k|\mathbf{z}_t,\theta,s]\ \log(p(\mathbf{z}_t,k|\theta',s)) \tag{5}$$

where θ represents our current estimates of the environmental parameters and θ' represents the new values we are searching.

Our algorithm has three basic steps, 1) transformation of the speaker dependent codebook vectors by $\mathbf{f}$, 2) use of the transformed vectors to compute accumulators (E step), 3) use of the acumulators to update θ (M step). After EM has converged, our algorithm maps the last transformed vectors $\{\mathbf{v}(k,s)\}$ into the representation used for the speaker recognizer. The likelihood that an utterance was generated by a particular speaker can then be measured with the AHS distance [2] or the maximum likelihood distance introduced by Gish [4]. Another possibility is to use the likelihood computed by the EM algorithm as the distance measure between an utterance and a speaker. In this case after EM converges no further computations are required.

To summarize, the whole process of speaker recognition follows these steps:

1. Training: by computing and storing the mel-frequency spectral (MFSC) vectors for a speaker's training utterance.
2. Estimation of environmental parameters: for each speaker and each utterance load the speaker's training features and run the DVTS algorithm against the utterance's MFSC features until the EM algorithm converges.
3. DVTS compensation of codebook: use the environmental parameters computed using the EM algorithm to map the speaker's training speech (this is actually done by the EM algorithm at each step) to the target environment.
4. Reestimation of speaker model: compute sufficient statistics for speaker recognition from the mapped features and use those to compute the likelihood that the utterance was produced by the speaker.

[1] Notice also that $\mathbf{B}$ is equal to $\nabla_{\mathbf{v}(k,s)}\mathbf{f}(\mathbf{v}(k,s),\mathbf{n},\mathbf{q})$

5. Speaker recognition/identification: select the most speaker with the highest likelihood as the utterance speaker.

4. Experiments

Our experiments were conducted on the TIMIT [6] and NTIMIT databases [5]. The TIMIT database is a carefully recorded clean database of 630 speakers. The NTIMIT database was constructed by passing the TIMIT database in a loop over telephone channels in the New York and New England area. Ten utterances were recorded from each speaker. The first 7 utterances were used for building a model of the speaker and the remaining 3 utterances were used for identification experiments. We used the first 50 female speakers from the database in our experiments. In general the female speakers are more confuseable than the male speakers.

Figure 1 shows five different test conditions for this algorithm. The accuracy of each algorithm is plotted against the signal-to-noise ratio (SNR) for white, Gaussian, noise that was artificially added to the test utterances. The sampling error in all the curves is approximately ± 0.05 in accuracy. All curves were generated using the Gish ML distance metric.

The top solid curve is for matched speaker identification (SID) performance. For this curve the training speech was measured at the given SNR. This is the highest performance that can be achieved with a given scoring algorithm, but is not practical for a real situation because the SNR is unknown and varying in a real application.

The bottom solid line is the performance when no compensation is applied and the models are trained on clean speech. The figure shows that noise levels above 30dB cause a rapid decrease in performance. Since typical office environments are 20-25dB this drop limits deployed speaker identification performance.

The next three lines show the performance of the DVTS algorithm applied with three different levels of prior information. The dash-dot line directly below the matched curve is the performance of the DVTS when the SNR is known. This shows that for SNR's 20dB and higher the DVTS algorithm achieves matched performance. At levels below 20dB the DVTS approximation diverges from matched performance. The approximation may be improved by removing the assumption of fixed covariance for each transformed vector at the cost of increased computation.

The dotted line just above the no compensation approach is the DVTS performance when a simple histogram based heuristic is used to estimate the environmental parameters. This heuristic is the starting point for the EM algorithm. The results for the EM algorithm is given by the dashed line abouve the heuristic line. This result shows that optimizing the environment over each speaker using EM causes a drop in performance from the prior level. This is due to the additional width that is effectively introduced into each speaker's distribution by the environmental estimation. However the improvement in performance over no compensation, shows that there is sufficent additional information in the measured features to estimate both the speaker and the environment. A more constrained speaker model may shrink the distance between the known DVTS level and the EM level.

Train	Test	Metric	Perform.
NTIMIT	NTIMIT	ML	47/150
TIMIT	NTIMIT	ML	18/150
TIMIT	NTIMIT	DVTS-EM	37/150

Table 1: Performance of DVTS with the EM algorithm in cross training conditions between TIMIT and NTIMIT.

A second matched and unmatched performance experiment was performed using the TIMIT and NTIMIT databases. In all cases 16 kHz speech data was used. Table 1 shows that the DVTS algorithm substantially improves performance in this real cross condition test. There is substantial miss-match in this experiment. The NTIMIT data has no speech data about 4 kHz and has substantial noise levels, whereas the TIMIT data has speech up to 8 kHz and has an SNR of 40 or higher.

5. Conclusions

This paper described our preliminary results in applying a new model-based approach to environmental compensation for speaker recognition. The DVTS approach can be used with many different speaker scoring algorithms because it works directly on the training speech to estimate training speech that is matched to the environment using EM. The paper showed that this approach can improve performance at SNR's less than 20dB where the ML distance metric is used as the final scoring algorithm. The paper also showed that the algorithm is able to decrease the substantial miss-match between TIMIT and NTIMIT.

In order to improve performance with this algorithm, more constrained speaker models may be necessary. This would lower the amount of variation induced by the environment compensation and may close the gap between the known environment case and the EM estimated case.

6. REFERENCES

1. A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. PhD thesis, CMU, Department of Electrical and Computer Engineering, 1990.
2. F. Bimbot and L. Mathan. Text-free speaker recognition using an arithmetic-harmonic sphericity measure. In *Eurospeech*, pages 169–172, 1993.
3. M. J. F. Gales. *Model-Based Techniques for Noise Robust Speech Recognition*. PhD thesis, University of Cambridge., 1995.
4. H. Gish and M. Schmidt. Text-independent speaker identification. *IEEE Signal Proc. Magazine*, Oct. 1994.
5. C. Jankowsky, A. Kalyanswamy, S. Basson, and J. Spitz. Ntimit: A phonetically ballanced, contin-

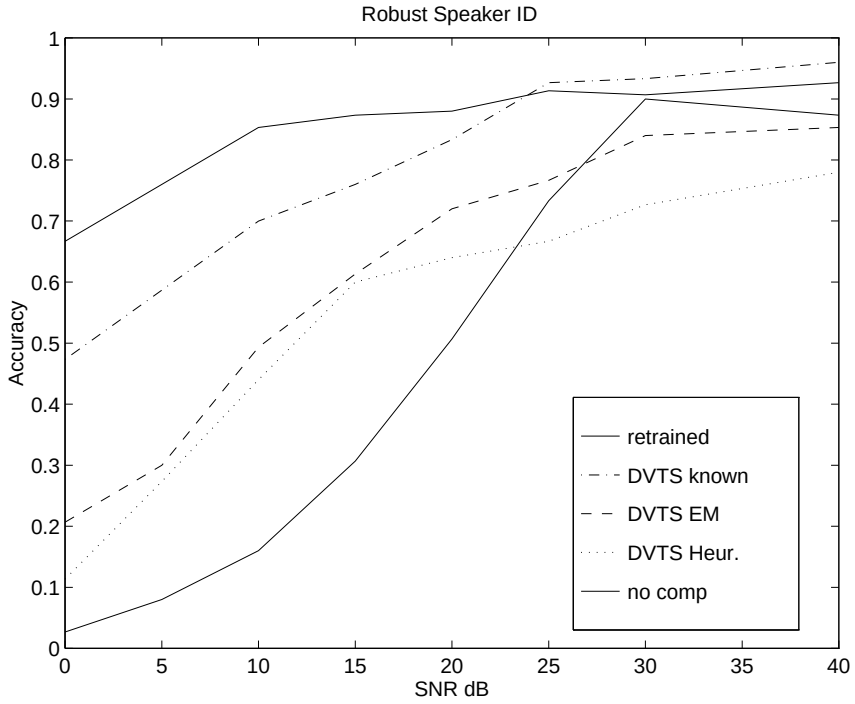

Figure 1: Performance of the DVTS algorithm on a subset of the TIMIT database as a function of the noise level.

uous speech, telephone bandwidth speech database. In *Proceedings: ICASSP 90. 1990 International Conference on Acoustics, Speech, and Signal Processing*, volume I, pages 109–112, April 1990.

6. L. Lamel, R. Kassel, and S. Senef. Speech database development: design and analysis of the acoustic-phonetic corpus. In *DARPA Speech Recognition Workshop SAIC-86/1546*, pages 100–109. Palo Alto, CA, February 1986.
7. C. J. Leggetter and P. C. Woodland. Speaker adaptation of hmms using linear regression. Technical Report CUED/F-INFENG/TR. 181, Cambridge University Engineering Dept., 1994.
8. P. J. Moreno. *Speech Recognition in Noisy Environments*. PhD thesis, CMU, Department of Electrical and Computer Engineering, 1996.
9. D. Reynolds. A gaussian mixture modelling approach to text-independent speaker identification. Technical Report Technical Report 967, Lincoln Lab, MIT, 1993.
10. X. Zhang and R. J. Mammone. Channel and noise normalization using affine transformed cepstrum. In *International Conference on Spoken Language Processing*, 1996.