EFFICIENT ON-LINE ACOUSTIC ENVIRONMENT ESTIMATION FOR FCDCN IN A CONTINUOUS SPEECH RECOGNITION SYSTEM

Jasha Droppo, Alex Acero, and Li Deng

Microsoft Research
One Microsoft Way, Redmond, Washington 98052, USA
{jdroppo,alexac,deng}@microsoft.com

ABSTRACT

There exists a number of cepstral de-noising algorithms which perform quite well when trained and tested under similar acoustic environments, but degrade quickly under mismatched conditions.

We present two key results that make these algorithms practical in real noise environments, with the ability to adapt to different acoustic environments over time. First, we show that it is possible to leverage the existing de-noising computations to estimate the acoustic environment on-line and in real time. Second, we show that it is not necessary to collect large amounts of training data in each environment–clean data with artificial mixing is sufficient.

When this new method is used as a pre-processing stage to a large vocabulary speech recognition system, it can be made robust to a wide variety of acoustic environments. With synthetic training data, we are able to reduce the word error rate by 27%.

1. INTRODUCTION

As speech recognition continues to move out of pristine laboratory environments and into real world recognition applications, noise robustness becomes a necessary component of any application. It is no longer safe to assume that speech input comes from a known microphone through a channel with high signal to noise ratio. Consequently, systems must be modified to deal with these harsher environments.

There is ongoing research into both feature-domain and model-domain techniques to improve the robustness of speech recognition systems. It has recently been shown [5] that, in a known environment, a feature-domain technique can achieve higher recognition accuracy than using matched noisy training and testing conditions. Since this matched condition is the limit that any model-domain technique strives for, we focus on feature-domain techniques that allow us to beat the limit.

One general method for feature-domain cepstral de-noising is to design a module that pre-processes cepstra before they are fed into a speech recognition system. This includes parametric feature space transformations [1, 2], spectral subtraction, VTS, CDCN [3], FCDCN [4] and most of its descendants, and cepstral smoothing techniques such as RASTA and CMN. The advantage of all of these techniques is that they can be seamlessly integrated into existing systems, without a complete overhaul of existing code.

Of these feature-domain cepstral pre-processing techniques, most have the same goal. Namely, to find the expected value of the clean cepstral vector, given the noisy observation. They tend to fall into two categories: those that parameterize the environment and then learn those parameters from the test data, and those that assume the noise environment is known and directly learn the transformation from noisy cepstra to clean cepstra.

The main advantage of adopting the parametric approach is obvious. It allows for adaptation to unseen, unknown, and slowly changing conditions. But there are disadvantages to the parametric approach that are wedded to its fundamental design. First, for tractability, it is fashionable to assume a linear distortion channel and stationary additive noise. This limits the system's ability to deal with nonlinear transducers, cross-frame effects, or nonstationary additive noise. Secondly, these systems tend to use gradient descent techniques to track the system parameters. As such, there is a tradeoff that needs to be made between how quickly the model can adapt to changing conditions and how precisely it can model stationary conditions.

The nonparametric approaches make no assumptions about the nature of corruption caused by the acoustic environment, whether stationary or not. They directly estimate the necessary transformation from noisy to clean cepstrum. FCDCN accomplishes this with a piecewise linear function which can be estimated to an arbitrary precision. It is arguably the simplest and most effective technique to use when the environment is known.

There are two disadvantages that have made FCDCN impractical in the past, and both are addressed in this paper.

First, if the testing acoustic environment does not match the training environment, recognition performance degrades. This is because FCDCN was designed to normalize one acoustic environment only. We show that it is practical to train multiple noise environment hypotheses to run in parallel, and choose among them at run time.

The second classic disadvantage is that it was believed that it was necessary to collect a large set of real stereo training data from the target environment. We show that a small amount of real noise synthetically mixed into a large, clean corpus is enough to achieve significant benefits.

Together, these two improvements allow us to build a FCDCN based continuous speech recognition system, with a reasonable amount of training data, that is robust to many types of noise.

Section 2 includes a conceptual overview of the FCDCN algorithm, and how stereo data is used to train the system for operation in a known acoustic environment. It is then shown in Section 3 how to make a maximum likelihood decision about the acoustic environment at run-time. This allows us to train parallel systems for exemplar acoustic conditions, and decide among them just prior to performing speech recognition. Section 4 discusses some results using this technique. In particular, we look at how the ML estimate can me smoothed over time, the frame-level accuracy of the environmental decision, and speech recognition performance.

2. REVIEW OF FCDCN

The parametric cepstral de-noising algorithms make assumptions about the structure of the corruption, usually a linear convolutional channel distortion and additive stationary noise. FCDCN, introduced in [4], bypasses issues associated with building a model of the corruption by directly estimating a function that maps from the corrupted signal space to the clean signal space.

For each noisy cepstrum $\mathbf{y}$, it computes the expected value of the clean cepstrum, $\mathbf{x}$ through

$$\hat{\mathbf{x}} = E\{\mathbf{x}|\mathbf{y}\}, \tag{1}$$

or equivalently,

$$\hat{\mathbf{x}} = \mathbf{y} + \mathbf{r}(\mathbf{y}), \tag{2}$$

where $\mathbf{r}(\mathbf{y})$ is the correction to apply at each point in the noisy cepstral space.

During training, the correction $\mathbf{r}(\mathbf{y})$ is approximated as a piecewise linear function. A vector quantization codebook is trained on the noisy cepstra $\mathbf{y}$, and for each codeword $\mathbf{c}_i$ the expected value of the difference between clean and noisy cepstra is stored as $\mathbf{r}_i$.

During testing, the index i of the codeword closest to the noisy cepstrum is found. Then, the correction vector $\mathbf{r}_i$ is added and denoising process is complete.

$$\hat{\mathbf{x}} = \mathbf{y} + \mathbf{r}_i \tag{3}$$

As a byproduct of the VQ search, distances are calculated between the noisy cepstrum and each codeword.

3. ENVIRONMENTAL ESTIMATION

Since FCDCN works well when the acoustic environment is known, an obvious extension is to train a rich set of FCDCN systems, and choose among them at run time. The problem that remains is finding an efficient and reliable way of making this choice.

This section shows how some calculations performed by the FCDCN algorithm can be re-cycled to estimate the probability of the acoustics given the environment, which then can be used to predict the most likely environment.

3.1. Bayesian Formulation

The conditional probability of the environment E given the acoustics A can be inferred using Bayes' rule and the implicit model of each acoustic space.

$$P(E|A) = P(A|E)\frac{P(E)}{P(A)} \tag{4}$$

It is safe to ignore $P(E)$ in Equation 4. If many frames are used to estimate $P(A|E)$, the relative importance of this prior will diminish. Also, since the marginal $P(A)$ is independent of the environment, and we are only interested in the most likely environment, it can be ignored.

The maximum likelihood estimate of the environment is then

$$\hat{E} = \text{argmax}_E P(A|E). \tag{5}$$

That is, to find the maximum likelihood environment, use the given acoustics to estimate $P(A|E)$ for each E, and then choose the environment that maximizes this quantity.

Fortunately, FCDCN can be modified to directly produce values that can be used as estimates of this conditional probability. In the past, codebooks have been trained on clean speech and the expected value of the difference between noisy and clean cepstra, for each codeword in the clean space [3].

If instead the base codebook is trained on the noisy data for each noise environment, it forms an implicit probability distribution function for the noisy cepstra given the noise environment. As part of the VQ search, distances are calculated between the current data frame and each codeword of each noise environment. Interpreting each codeword $\mathbf{c}_i^E$ as one mixture in a Gaussian mixture model, we can evaluate the conditional probability of the acoustics given the environment,

$$P(A|E) = \frac{1}{\sqrt{2\pi}} \sum_i \exp(-\frac{1}{2} d(\mathbf{y}, \mathbf{c}_i^E)) P(\mathbf{c}_i^E) \tag{6}$$

If we further assume a uniform prior on the codewords, $P(\mathbf{c}_i^E)$ can be ignored, and the conditional probability of the acoustics for a given noise environment is directly computable from the distortions calculated in the VQ processing.

Further shortcuts are also possible. It is reasonable to assume that the closest codeword dominates the calculation of $P(A|E)$, so that it can be approximated as

$$\hat{P}(A|E) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\min_i d(\mathbf{x}, \mathbf{c}_i^E)\right)\right), \tag{7}$$

which is a monotonic function of the minimum codeword distance.

3.2. On-Line Environmental Estimation

Essentially, reliable estimation of the environment can be made by examining the average codebook distortion between the test data and each known environment. In [6], it is shown that sentence-level decisions are reliable enough to reap the benefits of FCDCN without prior knowledge of the noise type. Unlike previous work, our noise adaptation is done on-line, that is, the estimate of the noise environment is updated on a frame-by-frame basis.

It is assumed that the testing conditions are close to one of the training conditions. In the worst case, one would expect the chosen FCDCN codebook to do no worse than leaving the signal unprocessed, because the nature of the transform is to try and move the noisy feature space to match the prior distribution of clean data.

One convenient side-effect of this de-noising system is that errors only occur when they are not important. That is, when the system becomes confused it is because the incoming noisy data matches more than one trained noise condition. In this case, the noise condition matches multiple conditions equally well and it shouldn't matter which decision is made–any one should be appropriate.

The $P(A|E)$ estimate is smoothed over time for each environment E in two stages. In the first stage, an FIR filter is applied identically to each estimate, with a time constant chosen to reduce the noise of the estimation while also allowing the estimates to adapt to changing conditions. At time n, for an environment E, this first smoothing produces

$$P_n^{(1)}(A|E) = \sum_{m=0}^{M-1} h[m] P_{n-m}(A|E). \tag{8}$$

The advantages of keeping this time constant short are clear–if someone using a mobile, speech-enabled device were to enter

an elevator or to emerge from the lobby of a quiet building into the pouring rain, the algorithm should be free to adapt to the new acoustic environment quickly. The filter was chosen to be rectangular with a length of 500ms. It is assumed that the acoustic environment does not change drastically at smaller time scales than this.

The linear smoothing is not always enough, so a second stage of smoothing consists of a moving-mode filter that constrains the rate of decision making. This non-linear filter examines the previous 64 frames and selects the environment with the highest likelihood in the most number of frames. As a result, there are fewer single and double frame errors in the environmental decision. The filter, while still allowing a fast decision transition, provides enough smoothing to get nearly perfect results in our tests.

4. RESULTS

The data for these experiments consists of three sets of training data and testing data, which correspond to three different acoustic environments. In general, the training data consisted of a clean Wall Street Journal (WSJ) corpus with synthetically mixed additive natural noise. The testing data consisted of 167 sentences of WSJ utterances with either synthetically mixed additive noise or real signals collected in a noisy environment.

The first set of data, labeled "Office" in our tests, was designed to approximate speech in a private office collected by a desktop microphone. Both the training and testing waveforms were corrupted by adding noise sampled separately from a desktop microphone.

The second set of data, labeled "Clean" in our tests, was designed to approximate speech into a closetalk microphone. The training and testing waveforms were unmodified from the original corpus.

the testing data consisted of real recordings

The third set of data was designed to approximate speech into a mobile device. For this set, only the training data was synthetically mixed. Ambient office noise was collected on a Compaq iPaq PocketPC and added to the clean corpus to create the training data. By contrast, the testing data consisted of real recordings collected on a similar device in a similar acoustic environment. The utterances were simultaneously using the device's built-in microphone and a closetalk microphone. This was not necessary for the cepstral de-noising, but enabled comparative recognition performance results.

4.1. Smoothed Conditional Probability Estimate

Figure 1 compares the conditional probability estimation before and after smoothing, for a single utterance with an associated noise condition that has been seen in the training set. The signal consists of a single word spoken into a closetalk microphone. The word begins at frame 30 and continues to frame 64.

To produce this figure, three codebooks were trained in advance for three acoustic environments. FCDCN was run in parallel for the three environments, and the codeword distortions were harvested to estimate the conditional probability $P(A|E)$.

The difference between the "Clean" condition and the "Office" condition is that the latter has additive office noise, so we would expect most of the discrimination to occur in regions with low SNR. This is indeed the case; the system separates these two conditions well in the absence of speech.

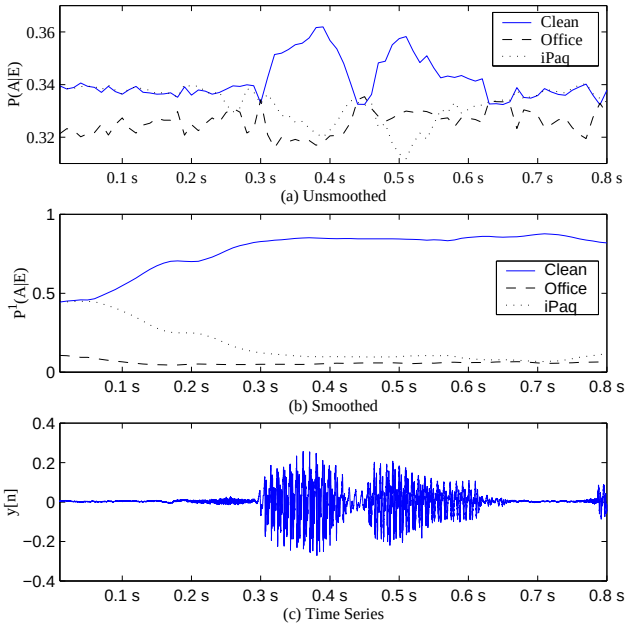

Fig. 1. Conditional probability measure before and after smoothing. (a) Normalized probability using one frame of codebook distortion. (b) Normalized probability using 32 frames of codebook distortion. (c) Corresponding time series.

The difference between the "iPaq" condition and the "Clean" condition is that the speech is corrupted by a different channel, but the additive noise is similar. Again, the discrimination occurs where we would expect, this time in the high SNR regions of the signal.

These two cases motivate the smoothing of the conditional probability estimates (Equation 8). To discriminate acoustic environments with dissimilar additive noise, one must smooth the estimate long enough to keep the discrimination information occurring between words. Conversely, to discriminate acoustic environments with dissimilar convolutional channels, we need to retain the conditional probability estimate across speech boundaries.

4.2. Misclassification Errors

Figure 2 shows the frame-level probability of error for the task of choosing among the three acoustical environments, as the length of the FIR smoothing filter increases. Even at a filter length of eight frames (80ms), less than five frames in 10,000 are misclassified with the chosen environments. This corresponds to an average of one error every twenty seconds, and has a negligible effect on recognition accuracy.

4.3. Robust Speech Recognition

Table 1 shows the recognition accuracy of this system across different training and testing scenarios. The testing data, a 167 sentence WSJ test set, was collected simultaneously on a hand-held device and on a closetalk microphone. The closetalk microphone data was not necessary for the cepstral de-noising, but was collected to produce a reasonable upper bound on the performance of the cepstral de-noising algorithm.

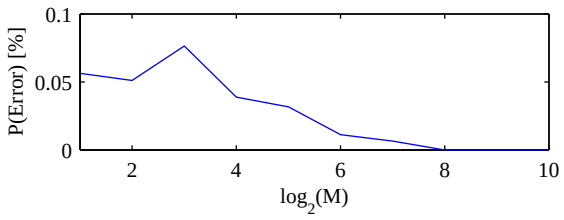

Fig. 2. Frame level probability of error

Microphone	Acoustic Model	Word Error Rate
Closetalk	Clean	6.76%
iPaq	Clean	10.40%
iPaq + FCDCN	Clean	8.86%
iPaq	iPaq (Artificial)	7.98%
iPaq + FCDCN	iPaq (Artificial) + FCDCN	7.61%

Table 1. Performance on real data

The baseline system uses a version of the Microsoft continuous density speech recognition engine (Whisper). The system uses 6000 senones (tied HMM states) and 20 Gaussians per state. The recognition task is 5000-word vocabulary, continuous speech recognition with a fixed bigram language model.

For reference, the first row of Table 1 shows the practical limit of any de-noising algorithm. When clean models are used to recognize data taken on the closetalk microphone, the accuracy is 93.24%.

The second row represents taking unmodified data from the hand-held device and recognizing it with models trained with clean WSJ data. Not surprisingly, this mismatched condition is the worst result. When the test data is de-noised with FCDCN, the accuracy increases by 1.5% absolute, which is a confirmation that the de-noising algorithm is modifying the data in the correct way.

It is a commonly held belief that using matched conditions in training and testing is the best one can do with noisy data. The fourth row of Table 1 shows that when we train models on data with utterances modified to simulate the noise picked up by the hand-held device, we do fairly well. But it is not the best we can do.

The last row of the table represents a different kind of matched condition. The only difference from the previous experiment is that both the training and the testing data have been passed through the cepstral de-noising algorithm. Similar results have been shown previously in [5, 7]. Even though conventional wisdom would indicate that the best one can do is to train models based on a matched noisy condition, in this case the matched de-noised condition reduces the word error rate by 4.7%.

Since reliable environment data is available from the cepstral de-noising algorithm, we can expect to approach this result closely. That is, even when the acoustic environment is unknown at runtime, we can clean the cepstrum and use a matched acoustic model to achieve optimum performance.

5. CONCLUSION

Cepstral de-noising algorithms which act independently of the recognizer on the cepstral stream can significantly improve the robustness of existing systems to additive noise and channel effects. One of the most successful algorithms, FCDCN, requires a knowledge of the current noise environment to operate. We have shown that the calculations inherent in the FCDCN algorithm can be leveraged to generate a real time, accurate estimate of the noise environment, making the system robust to both seen and some unseen conditions.

We have further shown that it is not necessary to obtain large amounts of speech in different acoustic environments. If ambient noise from the environment is available, synthetically mixed training data can be used to gain large improvements in recognition accuracy.

Used in conjunction with a sufficiently rich set of noise conditions and noise adaptive training, this cepstral de-noising algorithm can make continuous speech recognition robust to different acoustic environments [7].

6. REFERENCES

[1] Y. Ephraim and M. Rahim, "On second-order statistics and linear estimation of cepstral coefficients," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 162–176, March 1999.

[2] Yunxin Zhao, "Frequency-domain maximum likelihood estimation for automatic speech recognition in additive and convolutive noises," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 3, pp. 255–266, May 2000.

[3] Alex Acero and Richard M. Stern, "Environmental robustness in automatic speech recognition," in *Proceedings of the 1990 ICASSP*, April 1990, vol. 2, pp. 849–552.

[4] Alex Acero and Richard M. Stern, "Robust speech recognition by normalization of the acoustic space," in *Proceedings of the 1991 IEEE ICASSP*, April 1991, vol. 2, pp. 893–896.

[5] Li Deng, Alex Acero, Mike Plumpe, and X. D. Huang, "Large vocabulary speech recognition under adverse acoustic environments," in *Proceedings of the 2000 ICSLP*, Beijing, China, October 2000, pp. 806–809.

[6] Fu-Hua Liu, Richard M. Stern, Alex Acero, and Pedro J. Moreno, "Environment normalization for robust speech recognition using direct cepstral comparison," in *Proceedings of the 1994 IEEE ICASSP*, April 1994, vol. 2, pp. 61–64.

[7] Li Deng, Alex Acero, Li Jiang, J. Droppo, and Xuedong Huang, "High-performance robust speech recognition using stereo training data," in *Submitted to ICASSP 2001*, 2000.