FIGURES

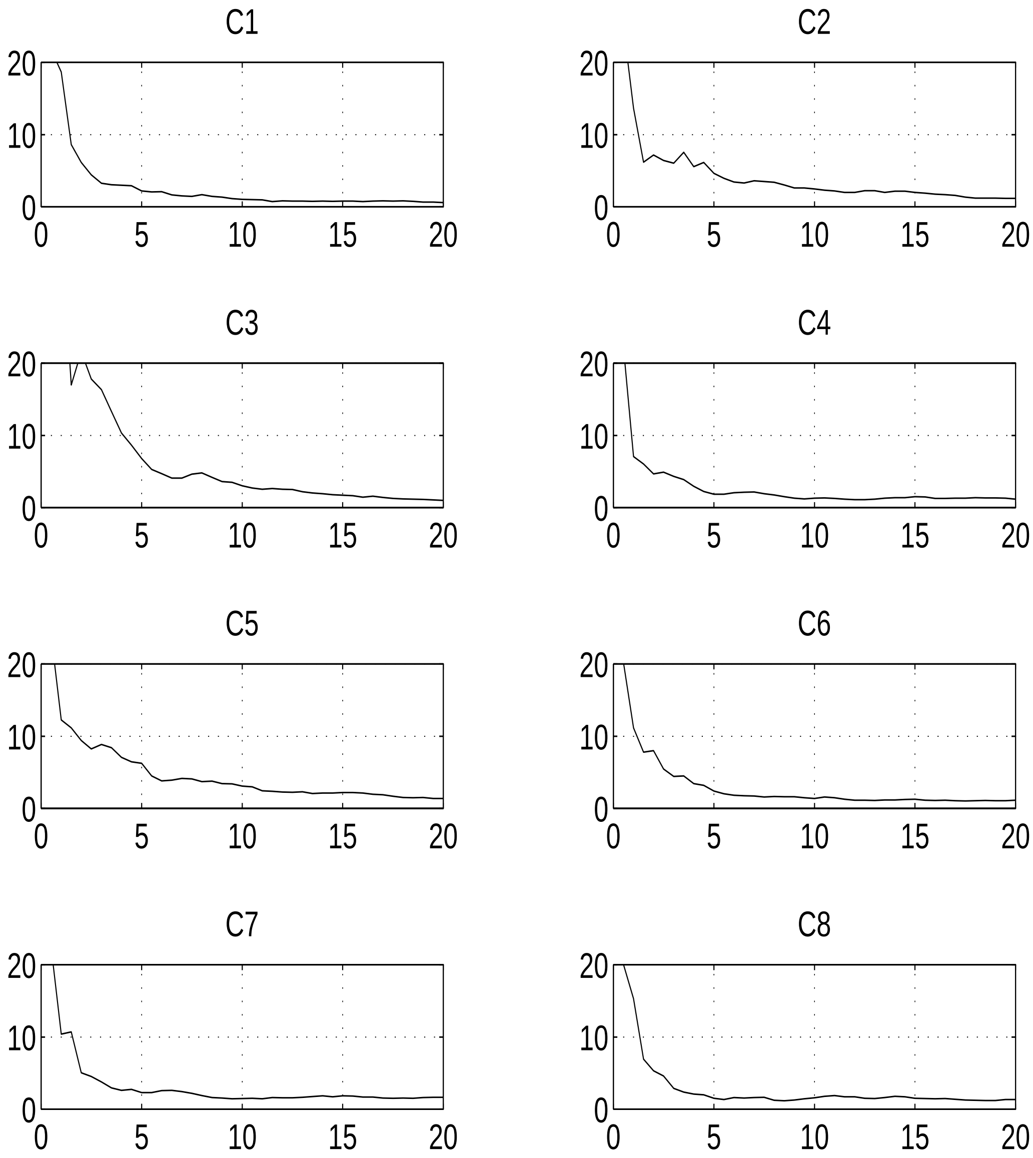

FIGURE 1. Average error in the channel estimate (as a percentage of the total variance) as a function of the estimation interval (in seconds) for cepstral coefficients C1 through C8

Description of the Experiment	**Train Data**	**Test Data**	**Word Error (%)**
Baseline	CC	CC	**68.1**
Cross-Database	WSJ	CC	**71.5**
Cross-Database in Noisy Data	WSJ	nCC	**78.9**
Adaptation of WSJ Gaussian Mixtures	WSJ/CC	CC	**69.7**
CC Booted from WSJ Models	WSJ/CC	CC	**67.1**

TABLE 8. Summary of cross-database acoustic training results on the credit card task

c1	*c2*	*c3*	*c4*	*c5*	*c6*	*c7*	*c8*	*c9*	*c10*	*c11*	*c12*
71.4	79.8	60.1	70.8	49.5	71.4	53.5	78.0	68.0	51.1	25.1	57.9

TABLE 5. Inter-speaker variance of the cepstral mean measurements as a percentage of the total variance

Parameter	HQ	TQ
Sampling Rate	16 kHz	8 kHz
Number of FFT Coefficients	256	128
Number of Cepstral Coefficients	12	8
Number of Filters	25	18
Total Bandwidth	100-6400 Hz	300-3000 Hz

TABLE 6. Parameters used in the high-quality (HQ) and telephone-quality (TQ) front ends

Acoustic Model Training		Test Set Word Error (%)	
Training Data	**Signal Processing**	**Sennheiser (HQ data)**	**Telephone (TQ data)**
Sennheiser (HQ)	High-Quality Front End	7.8	19.4
Sennheiser (HQ)	Telephone Front End	9.0	9.7
Telephone (TQ)	Telephone Front End	10.0	10.3
Sennheiser (HQ)	Telephone Front End without Cepstral-Mean Normalization	9.4	11.2

TABLE 7. Effect of different training and signal processing on test set performance.[1]

[1] **Results are word error rate on the 400-sentence simultaneous test set.**

Mic Type	Microphone Description
A	Radio Shack Pro-Unidirectional Highball 33-984
B	Sony ECM-55
C	Sony ECM-50PS
D	Crown PCC-160 Phase-Coherent Table-Top
E	Shure SM91 Unidirectional Condenser
F	AT&T 720 Handset with Speech over Local Telephone Lines
G	AT&T 720 Speaker Phone with Speech over Local Telephone Lines
H	Crown PZM-6FS Pressure Zone Table-Top

TABLE 2. Listing of microphone types in development test set

Algorithm	Sennheiser Microphone	Other Microphone
Cepstral Mean Removal	14.5	22.8
DFT Equalization	14.4	22.6

TABLE 3. WSJ 5K NVP Development test set word error rate

Algorithm	Word Error Rate (%)
CMN	21.6
Channel Estimation	21.4

TABLE 4. Channel equalization results on WSJ development test set

TABLES

Spkr Index	Senn Error Rate	OMic Error Rate	Error Ratio (OMic/ Senn)	Mic	Relative Distortion						
					Cep	D Cep	DD Cep	Egy	D Egy	DD Egy	Avg
426	7.3	5.2	**0.7**	A	0.60	0.57	0.60	0.23	0.19	0.20	**0.40**
22h	6.3	8.0	**1.3**	B	0.56	0.58	0.61	0.40	0.41	0.44	**0.50**
22k	12.5	16.8	**1.3**	B	0.48	0.54	0.58	0.34	0.27	0.30	**0.42**
052	9.0	10.4	**1.2**	C	0.65	0.66	0.68	0.73	0.55	0.57	**0.64**
061	8.2	11.0	**1.3**	C	0.59	0.62	0.65	0.65	0.50	0.53	**0.59**
00b	15.7	24.8	**1.6**	C	0.60	0.61	0.63	0.68	0.47	0.50	**0.58**
001	5.6	6.9	**1.2**	D	0.62	0.59	0.61	0.58	0.43	0.45	**0.55**
00d	21.0	34.5	**1.6**	D	0.72	0.73	0.77	0.49	0.31	0.32	**0.56**
22l	10.4	17.2	**1.7**	D	0.58	0.62	0.65	0.53	0.47	0.50	**0.56**
22g	6.7	11.9	**1.8**	D	0.62	0.68	0.72	0.60	0.51	0.54	**0.61**
431	17.7	32.5	**1.8**	E	0.63	0.65	0.67	0.70	0.50	0.51	**0.61**
422	20.9	40.1	**1.9**	F	0.92	0.81	0.82	0.38	0.31	0.33	**0.60**
400	13.8	30.7	**2.2**	G	0.83	0.81	0.83	0.53	0.61	0.65	**0.71**
423	9.6	24.8	**2.6**	G	1.00	0.87	0.87	0.43	0.50	0.55	**0.70**
424	12.3	32.0	**2.6**	G	0.99	0.90	0.92	0.52	0.63	0.68	**0.77**
00c	16.5	38.5	**2.3**	H	0.78	0.79	0.82	1.14	0.74	0.76	**0.84**
051	8.3	23.1	**2.8**	H	0.80	0.86	0.90	1.20	0.69	0.72	**0.86**
060	8.7	24.8	**2.9**	H	0.76	0.77	0.79	0.97	0.66	0.69	**0.77**
Avg	11.7	21.8	**1.8**		0.71	0.70	0.73	0.62	0.49	0.51	**0.63**

TABLE 1. Error rate and distortion for 18 WSJ0 development test speakers

[11] MADCOW, “Multi-Site Data Collection for a Spoken Language Corpus, *1992 DARPA Speech and Natural Language Workshop Proceedings*, pp. 7-14.

[12] J.J. Godfrey, E.C. Holliman, and J.McDaniel, “SWITCHBOARD: Telephone Speech Corpus for Research and Development,” in Proc. *IEEE ICASSP*-92, pp. I-517-I-520.

[13] L.E. Baum, T. Petrie, G. Soules and N. Weiss, “A Maximization Technique in the Statistical Analysis of Probabilistic Functions of Finite State Markov Chains,” in *Ann. Math. Stat*., Vol.41, pp. 164-171, 1970.

REFERENCES

[1] G. Doddington, “CSR Corpus Development,” in *DARPA SLS Workshop*, Feb 1992.

[2] S.F. Furui, “Cepstral Analysis Technique for Automatic Speaker Verification,” in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 254-272, Apr. 1981.

[3] H.L. Van Trees, *Detection, Estimation, and Modulation Theory,* John Wiley & Sons, 1968.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum Likelihood Estimation from Incomplete Data,” in *Journal of the Royal Statistical Society (B),* Vol. 39. No. 1., pp. 1-38, 1977.

[5] L.R. Rabiner, J. G. Wilpon, and B.H. Juang, “A Segmental K-means Training Procedure for Connected Word Recognition,” in *AT&T Technical Journal*, pp. 21-40, May-June 1986.

[6] V. Digalakis, H. Murveit, “Genones: Optimizing the Degree of Mixture-Tying in a Large-Vocabulary HMM-Based Speech Recognizer,” *Proc. ICASSP,* pp. I-537 - I-540, April 1994.

[7] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, “Large Vocabulary Dictation using SRI’s DECIPHER^{TM} Speech Recognition System: Progressive Search Techniques,” *Proc. ICASSP,* pp. II-319 - II-322, April 1993.

[8] S.B. Davis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357-366, Aug. 1980.

[9] M. Weintraub and L. Neumeyer, “Constructing Telephone Acoustic Models from a High Quality Speech Corpus,” *Proc. ICASSP*, pp. I-85 - I-88, April 1994.

[10] L. Neumeyer and M. Weintraub, “Probabilistic Optimum Filtering for Robust Speech Recognition,” *Proc. ICASSP*, pp. I-417 - I-420, April 1994.

4 SUMMARY

To compensate for channel and microphone mismatch we investigated the validity of two simplifying assumptions of the popular cepstral-mean normalization algorithm. To remove these assumptions, we introduced two new channel normalization algorithms. Our experimental results showed that on the WSJ alternate-microphone task the cepstral-mean normalization algorithm was as effective as the proposed channel normalization algorithms.

We also presented our approach to developing acoustic models for telephone applications. We showed that we can take advantage of existing, “high-quality” data and achieve similar performance with cross-database training to that obtained using task-dependent training.

To test our ideas on the CC task we decided to train the acoustic models using 7,000 WSJ sentences. For the CC task, training the models with WSJ data presents mismatches along a number of dimensions, which include:

- Acoustics of recording (high-quality versus telephone)
- Vocabulary independence (WSJ does not have the same focus as the credit card conversations)
- Amount of training data (WSJ has 7000 training sentences, CC has 1000)
- Speaking modes (read versus spontaneous speech)

We ran the recognition experiments using SRI’s DECIPHER ™ phonetically-tied mixture system with a TQ front end. All the recognition experiments are gender-dependent, use a bigram grammar, and are expressed in terms of word error rate. The test consisted of 167 sentences. The results are summarized in Table 8. In the baseline experiment, where we trained and tested the models using CC data, the error rate was 68%. The cross-database experiment yielded a slightly higher error of 71.5%. We also tested the WSJ-trained models with a noisy version of the test set (nCC). The data was corrupted with mid-continental US voice channel effects and highway noise recorded in the interior of a Ford Taurus on the highway. The average signal-to-noise ratio after adding the noise was 20 dB. The error for the nCC test set was 78.9%.

To improve performance in the cross-database experiment we adapted the distributions of the HMM using the CC train set. To adapt the models we reestimated the parameters of the Gaussian distributions (means and variances) using the forward-backward algorithm [13]. The mixture weights and state transition probabilities remained unchanged. This approach reduced the error to 69.7%. Finally, we ran two additional iterations of the forward-backward algorithm on the WSJ-trained models using the CC train set. This run produced the best result of 67.1% error rate.

The cross-database results are very close to the baseline despite the mismatches between the two databases. Based on previous experiments, we believe that the difference in the results is more likely to be caused by mismatches in speaking modes and vocabulary than in the acoustics of the recording environment.

used (9.0% and 9.7% error respectively). Here the robustness of the recognizer is increased at the expense of performance in the HQ test condition. The next line in the table shows that training the models with TQ data actually degrades performance even for the TQ test condition (10.0% and 10.3% for HQ test and TQ test conditions). This is an important result since it indicates that we can train TQ models using HQ data with no degradation in performance. This is no longer true when we eliminate the cepstral-mean normalization (CMN) algorithm [2] as shown in the last line of the table. This degradation in performance is caused by the stationary convolutional noise (9.4% and 11.2% for HQ test and TQ test conditions when CMN is not used).

In summary, we can train the recognizer models using a telephone bandwidth front end and high-quality training data. The drawback of the method, however, is that separate models have to be trained for HQ and TQ applications. Another limitation of this experiment is that all the telephone data were recorded using the same local telephone line. Therefore we cannot predict from these experiments on a small stereo speech corpus how the variability of a wider telephone network will affect the recognition performance. For this reason, we test telephone models trained with HQ data on a more realistic database: the Switchboard speech corpus.

3.3 Experimental Results on the Switchboard Corpus

In this experiment we also show how to train HMM models for OTP applications using a HQ database and how they compare to models trained with TQ data. The test is performed on the *Credit-Card* (CC) task that is part of the *Switchboard* [12] speech corpus, a large speech database recorded over the public telephone network. For training we use the WSJ database that was recorded using high-quality Sennheiser microphones. The CC corpus consists of spontaneous telephone conversations between two individuals talking about issues related to credit cards. In contrast, the WSJ corpus was recorded from subjects reading sentences extracted from the *Wall Street Journal* newspaper.

We will focus on the design of robust features for OTP applications by using a standard filterbank-based front end [8] tuned for telephone-bandwidth applications. In Table 6 we show the parameters used in our wide-bandwidth (HQ) and telephone-bandwidth (TQ) front ends[5]. The main difference in the signal analysis stage is the total bandwidth of the filterbank. Both front-end signal processing modules produce six feature streams: cepstral energy (C0), cepstrum and their first- and second-order differences. The mean of each cepstral coefficient is removed on a per-sentence basis.

3.2 Experimental Results on the ATIS Corpus

We have considered some of the approaches mentioned in Section 3.1 in the past [9][10] and found that an adequate front end can minimize the mismatch between the acoustic spaces. In fact, in a pilot study conducted at SRI [9], we found that the variability introduced by the telephone handsets had little effect on recognition performance.

For our pilot study, we collected a corpus of both training and testing speech using simultaneous recordings made from subjects wearing a Sennheiser HMD 414 microphone and holding a telephone handset. The speech from the handset was sent over local telephone lines. Ten different handsets were used by 13 male subjects (10 for training and 3 for testing) who read ATIS (Air Travel Information System) sentences [11]. The selected telephones included three carbon button, two inexpensive Radio-Shack, and a variety of telephones found in our lab. The amount of data was 3,000 sentences for training and 400 sentences for testing.

Table 7 shows the results for different training and testing conditions. When the models are trained with HQ data and the HQ front end is used to generate the features we get the best possible result in the train HQ/test HQ condition (7.8% word error rate) and the worst result when we test on the TQ data (19.4%). This shows how the error is doubled due to the mismatch in the higher frequencies of the spectrum. The difference in error rate between the test HQ and test TQ conditions is greatly reduced when the TQ front end is

[5] We shall use HQ/TQ to denote both the high-/telephone-quality data and the wide-/telephone-bandwidth front end, respectively.

to obtain the channel estimate has on the accuracy of the estimate. As usual, we assumed that the channel does not vary within each speaker’s 20-sentence section. Under this assumption, we can accurately estimate the channel response in the cepstral domain by computing the average of each cepstral coefficient over the whole 20-sentence section. We can then use this channel estimate to compute the average error in the less accurate channel estimates that are obtained using shorter intervals. In Figure 1 we have plotted the error in the channel estimate as a percentage of the total variance of the corresponding cepstral coefficient and as a function of the estimation interval’s length. The plots are averaged over all the intervals, sentences and speakers. We can see that, for an estimation interval of 8 seconds, the estimation error is small, and varies from 1.2% to 4.8% for different cepstral coefficients. The average estimation error over all cepstral coefficients for 8-second long intervals is 2.5%.

3 TRAINING ISSUES

3.1 Construction of Telephone-bandwidth Acoustic Models

Our objective is to train an HMM recognizer for over the telephone (OTP) applications without collecting specific training data for each task. For example, we would like to use available large speech corpora recorded with high-quality (HQ) microphones instead of collecting data over the telephone network. Here we show that the variability in the acoustics of the telephone quality (TQ) recordings has little impact on performance as long as: (1) cepstral mean normalization is used to compensate for channel variations, and (2) the signal analysis matches the spectrum of the telephone channel.

To avoid collecting new training data for a task in which there is a mismatch between training and test conditions, there are a number of possible approaches:

- Design robust features that are not affected by the variations in the microphone, background noise, channel distortion, and so forth.
- Adapt the parameters of the acoustic models.
- Map features between the test and train acoustic spaces. This means that we make the data used for testing look like the data used for training.

the serious channel mismatch between the Sennheiser recordings and the secondary-microphone recordings in the WSJ corpus, the results were essentially the same (21.6% with CMN and 21.4% with the proposed channel estimation algorithm). This indicates that the underlying assumption that $(1/N)\sum \boldsymbol{x}_n$ is independent of the sequence of distributions that generated $\boldsymbol{x}_n$ is fairly accurate for these long sentences (~8 seconds).

To test this hypothesis, we must compare for each speaker and channel the variation in the measurements of $(1/N)\sum \boldsymbol{x}_n$ when the transcription is fixed to the variation in the measurements of the same quantity when the transcription varies. To perform this comparison, we have to collect multiple recordings of each transcription for each speaker/channel combination. Assuming that the channel characteristics do not vary over the different recordings for a particular speaker/channel combination, we can then measure the cepstral mean for each sentence and group these measurements into sets based on the sentence transcription. Our hypothesis is then equivalent to the hypothesis that for each speaker/channel the averages of the cepstral mean values of the different groups are equal.

Since we did not have data to test this hypothesis directly, we measured the cepstral mean values for all 360 sentences in the test set. The variability in these measurements consists of two terms: the variability in the speaker/channel-dependent measurement of the channel h and the variability in the measurement of $(1/N)\sum \boldsymbol{x}_n$ (see (8)). Assuming that the channel characteristics do not vary during the 20-sentence section of each of the 18 speakers, then we can estimate each one of these two sources of variability by comparing the variance of the cepstral mean measurements within each 20-sentence section to the total variance. The results of these measurements for all 12 cepstral coefficients are presented in Table 5, where we show the inter-speaker squared error as a percentage of the total squared error. We can see that the inter-speaker variance represents the larger amount of the total variance for most cepstral coefficients.

This result agrees with our experimental finding that for the long WSJ sentences a satisfactory estimate of the channel can be obtained using CMN. Hence, we decided to perform an additional experiment to investigate the effect that the length of the interval used

For each speaker, the word error rate[4] is given in Table 1 for the Sennheiser channel as well as the secondary microphone channel (denoted OMic for Other Microphone). The ratio of these word-error rates is shown in the fourth column. The normalized mean-squared error distortion between the Sennheiser and the secondary microphone features was computed for each of the six features. They are listed in subsequent columns, followed by an average of all six distortions. Note that the word-error rate and the average distortion are fairly constant across speakers for a given OMic condition. The results, presented in Table 3, show that CMN is as effective as the LDMN equalization algorithm. To explain this result, we can either assume that the variation of convolutional noise within a spectral band is negligible, or that there are other factors that swamp its effects on recognition performance.

In a second experiment we compared the joint channel and model estimation algorithm to CMN on the same database. The joint channel/model estimation algorithm was implemented as follows. At each iteration during training, the most likely state sequence was estimated for each utterance in the training set. Equalization was performed in the cepstral domain: a separate estimate of the channel response was obtained for each utterance using (10), and subsequently subtracted from the cepstral vectors. Compensation was followed by an iteration of the forward-backward algorithm. We computed a total of two iterations of the sequential EM algorithm during training. During recognition, an initial estimate of the channel was obtained using CMN. The most likely state sequence was obtained from the Viterbi alignment of a first recognition pass, and a more accurate estimate of the channel response was found using (10). A second recognition pass was then performed after subtracting the new channel estimate from the cepstral vectors.

The results are summarized in Table 4. In this experiment we used a phonetically-tied mixture system—that is, it had a smaller degree of mixture sharing than the tied-mixture system used in the first experiment. In this system, all context-dependent models with the same center phone use the same mixture components in their output distributions. Despite

[4] The average word error rate in Table 1 is slightly different than the one showed in Table 3 because they have been computed with different training procedures.

mate based on *a-priori* knowledge is used to obtain the most likely state sequence. This state sequence can then be used to refine the channel estimate using (10), and the procedure can be iterated.

2.3 Experimental Results

To compare both normalization algorithms presented in Section 2.1 and Section 2.2 to the conventional CMN algorithm, we tested the algorithms using SRI's DECIPHER ™ continuous speech recognition system [6][7] on the 5,000-word alternate microphone task of the WSJ corpus. The system is configured with a six-feature front end that outputs 12 cepstral coefficients, cepstral energy, and their first- and second-order differences. The cepstral features are computed from an FFT filterbank. We used genonic hidden Markov models that allow an arbitrary degree of Gaussian sharing across different HMM states as described in [6]. For fast experimentation, we used the progressive search framework [7]: an initial recognizer with a bigram language model outputs word lattices for all the utterances in the test set. These word lattices are then rescored using our channel normalization algorithms. The models were trained using the large-vocabulary WSJ corpus recorded with a close-talking Sennheiser[3] microphone from male speakers. For testing we used a test set with simultaneous recordings. One channel contains speech recorded with the Sennheiser microphone, and the other channel was recorded using 8 different low-quality microphones and telephone handsets. There were 18 male speakers in the test set. Each speaker recorded 20 sentences, for a total of 360 sentences. In Table 1, the different speakers are grouped by secondary microphone type. The secondary microphone types are listed in Table 2. We first compared the LDMN algorithm to the conventional CMN. In this experiment we used a tied-mixture HMM system, with all HMM states sharing the same mixture components.

[3] All product names used in this paper are the trademark of their respective holders.

For fixed HMM parameters, (14) also guarantees that the likelihood does not decrease,

$$\log p(\boldsymbol{Y} \mid \theta_N, \boldsymbol{h}_N) \geq \log p(\boldsymbol{Y} \mid \theta_N, \boldsymbol{h}_O) . \tag{16}$$

Therefore, every combined iteration of (13) and (14) guarantees that the likelihood $p(\boldsymbol{Y} \mid \theta, \boldsymbol{h})$ does not decrease. For simplicity, however, and if we assume that the most likely state sequence is dominant [5], we can replace (14) by

$$\boldsymbol{h}_N = \underset{\boldsymbol{h}}{\operatorname{argmax}}\ p(\boldsymbol{Y} \mid S, \theta_N, \boldsymbol{h}) \tag{17}$$

and the channel estimate above can be computed using (10).

Recognition

In recognition, we want to determine the most likely state sequence. This implies that we should jointly maximize over the state sequence and the channel

$$\max_{S, \boldsymbol{h}} \quad p(\boldsymbol{Y}, S \mid \theta, \boldsymbol{h}) \tag{18}$$

and the maximization above can be performed by an alternation between maximizing over the state sequence and over the channel estimate, which is similar to the training algorithm described in the previous section.

To summarize, we presented algorithms that jointly estimate the HMM parameters and the channel during training, and the most likely state sequence and the channel during recognition. During training, we assume that the training data can be split into blocks and that the channel characteristics do not vary with time within each block. These blocks can be either single utterances, or sessions with multiple utterances. A single estimate of the channel response in the cepstral domain is estimated for each block. The training algorithm alternates between estimating the channel response and using the new channel estimate to obtain refined estimates for the HMM parameters. Hence, the output distributions directly model the cepstrum of the clean signal. During recognition, an initial channel esti-

Training

When the state sequence is not given, then one can use the Expectation-Maximization (EM) algorithm [4] to jointly estimate the channel and the HMM parameters by maximizing at each iteration the objective function

$$(\theta_N, \boldsymbol{h}_N) = \underset{\theta, \boldsymbol{h}}{\operatorname{argmax}} \quad E\{\log p(\boldsymbol{Y}, S \mid \theta, \boldsymbol{h}) \mid \boldsymbol{Y}, \theta_O, \boldsymbol{h}_O\} \tag{12}$$

where θ_O and $\boldsymbol{h_O}$ are the parameters from the previous iteration and, θ_N and $\boldsymbol{h_N}$ are the reestimated parameters.

The solution to the maximization problem above is fairly complex, however, and the channel and model estimates can alternatively be obtained by an iterative procedure, where one alternates between obtaining estimates of the model parameters and the most likely state sequence, and using these estimates to compute the estimate for the stationary channel. Each iteration of the algorithm is therefore broken down into two steps:

1. Using the previous channel estimate $\boldsymbol{h}_O$, reestimate the model parameters using a nested EM procedure:

$$\theta_N = \underset{\theta}{\operatorname{argmax}} \ E\{\log p(\boldsymbol{Y}, S \mid \theta, \boldsymbol{h}_O) \mid \boldsymbol{Y}, \theta_O, \boldsymbol{h}_O\} \tag{13}$$

where S denotes the most likely state sequence using the current model and channel estimate.

2. Obtain a new channel estimate by maximizing the likelihood of the observations given the newly obtained model parameters θ_N:

$$\boldsymbol{h}_N = \underset{\boldsymbol{h}}{\operatorname{argmax}} \ p(\boldsymbol{Y} \mid \theta_N, \boldsymbol{h}) \tag{14}$$

The EM procedure described in (13) is guaranteed that the likelihood will not decrease for a fixed channel estimate, that is

$$\log p(\boldsymbol{Y} \mid \theta_N, \boldsymbol{h}_O) \geq \log p(\boldsymbol{Y} \mid \theta_O, \boldsymbol{h}_O) \tag{15}$$

this introduces an error in the true speech cepstrum, which may lead to recognition errors. A better approach is to try to jointly estimate the channel $\boldsymbol{h}$ and the HMM parameters during training, and the channel and the state sequence during recognition.

Let us first assume that the HMM state sequence $[s_n]$, $n = 0, \ldots, N\text{-}1$ is given. Then, the maximum-likelihood channel estimate is given by

$$\hat{\boldsymbol{h}} = \underset{\boldsymbol{h}}{\operatorname{argmax}}\ p(\boldsymbol{Y} \mid S, \theta, \boldsymbol{h}) \tag{9}$$

where $\boldsymbol{Y}$ is the collection of observations, S is the state sequence, θ are the HMM parameters and $\boldsymbol{h}$ is the channel. For Gaussian output distributions, it can be shown [3] that this estimate is given by

$$\hat{\boldsymbol{h}} = \left[\sum_n (\boldsymbol{C}(s_n))^{-1}\right]^{-1} \sum_n (\boldsymbol{C}(s_n))^{-1} (\boldsymbol{y}_n - \mu(s_n)) \tag{10}$$

where the HMM output distribution

$$p(\boldsymbol{x}_n \mid s_n) = \mathcal{N}(\mu(s_n)\ ;\boldsymbol{C}(s_n)) \tag{11}$$

is a multivariate normal distribution with a state dependent mean $\mu(s_n)$ and covariance $\boldsymbol{C}(s_n)$. Hence, when the state HMM sequence is given, the channel estimate $\hat{\boldsymbol{h}}$ can be obtained as a weighted combination of the deviations of the observed features from the means of the HMM output distributions that are specified by that state sequence. The weights depend on the covariances of these output distributions. For HMMs with continuous mixtures as output distributions, (10) can be applied when both the state and the mixture index are known.

Below we examine how this channel estimate can be incorporated in the training and recognition problems.

where $\hat{X}_{k,n}$ is the equalized DFT energy, N is the number of frames in the sentence, and Q_k is the equalization factor for the k-th DFT energy component in the current sentence. With this algorithm we can eliminate the stationary convolutional noise in the sentence without the assumption that H_l is constant within the spectral band.

2.2 Joint Channel and Model Estimation

Using CMN to perform channel equalization is tantamount to the underlying assumption that the sample cepstral average of the “clean” signal is an invariant quantity. This assumption is clearly violated when CMN is used to estimate the channel in short utterances. We present a different approach for jointly estimating the channel and the HMM parameters during training, and for obtaining the channel and the most likely state sequence during recognition.

In the cepstral domain, the observed speech signal corrupted by stationary convolutional noise can be written as

$$\boldsymbol{y}_n = \boldsymbol{h} + \boldsymbol{x}_n \tag{7}$$

where $\boldsymbol{h}$ is the cepstrum of the channel response, x_n is the clean speech cepstrum at each frame $n = 0, \ldots, N\text{-}1$ in the sentence, and we assume that the channel characteristics do not vary with time over a single sentence. In CMN the estimated channel $\hat{\boldsymbol{h}}$ is computed as a time average of all the frames in the sentence

$$\hat{\boldsymbol{h}} = \frac{1}{N}\sum_{n=0}^{N-1}\boldsymbol{y}_n = \boldsymbol{h} + \frac{1}{N}\sum_{n=0}^{N-1}\boldsymbol{x}_n \tag{8}$$

If we assume that the sequence $\boldsymbol{x}_n$ is modeled using HMMs with Gaussian observation distributions, then CMN will give an unbiased estimate of $\boldsymbol{h}$ only when $(1/N)\sum\boldsymbol{x}_n$ is zero, or more generally, independent of the sequence of distributions that generated $\boldsymbol{x}_n$.

In practice, the above average will not be constant since it depends on the sequence of distributions that generated $\boldsymbol{x}_n$, that is, on the transcription of the sentence. The CMN algorithm will interpret these fluctuations as channel variations, and remove them. In effect, Consider the following speech signal corrupted with stationary convolutional noise,

$$y[t] = x[t] \otimes h[t] \tag{1}$$

where $x[t]$ is the clean speech sequence, $h[t]$ is the impulse response of the channel, and $y[t]$ is the distorted speech. After applying the Discrete Fourier Transform to a frame[2] of speech, we get the spectral energy equation,

$$Y_{k,n} = X_{k,n}H_k \tag{2}$$

where k is the DFT index and n is the frame index. The log filterbank energy is given by

$$\log F_{l,n} = \log \sum_k w_{k,l} X_{k,n} H_k \tag{3}$$

where $F_{l,n}$ is the filterbank energy for band l in frame n and $w_{k,l}$ is a filter weight coefficient (this coefficient is zero outside the spectral band of the filter). If we assume that H_k is constant within the frequency band l,

$$H_k = \tilde{H}_l \quad \forall k \colon w_{k,l} \neq 0 \tag{4}$$

we can express the log filterbank energy as follows:

$$\log F_{l,n} \cong \log\left(\tilde{H}_l \sum_k w_{k,l} X_{k,n}\right) = \log \tilde{H}_l + \log \sum_k w_{k,l} X_{k,n} \tag{5}$$

and the constant term $\log \tilde{H}_l$ is eliminated with cepstral mean subtraction.

To avoid the approximation in (4), we can simply normalize the spectrum in the log-DFT domain before the filterbank integration as follows:

$$\hat{X}_{k,n} = \exp(\log Y_{k,n} - \frac{1}{N}\sum_{m=0}^{N-1} \log Y_{k,m}) = \frac{Y_{k,n}}{\exp(\frac{1}{N}\sum_{m=0}^{N-1} \log Y_{k,m})} = \frac{Y_{k,n}}{Q_k} \tag{6}$$

[2] The waveform is subdivided in a sequence of overlapping segments called frames, usually at intervals of 10-20 ms. Each frame is windowed before computing the DFT.

telephone-bandwidth speech, we are able to achieve with cross-database training similar performance to task-dependent training.

The remainder of this paper is organized as follows. In Section 2, we explore two different channel normalization algorithms. The first algorithm performs cepstral normalization in the log-DFT domain rather than in the log-filterbank domain. The second algorithm jointly estimates the channel and the HMM parameters during training, and the channel and most likely HMM state sequence during recognition. The performance of these two equalization algorithms is similar to the cepstral-mean removal algorithm on the alternate-microphone task of the *Wall Street Journal* (WSJ) corpus [1]. In Section 3, we discuss techniques to train acoustic models with data recorded with a high-quality Sennheiser microphone for use over the telephone.

2 CHANNEL EQUALIZATION

Although cepstral-mean normalization (CMN) is a simple technique that has been effectively used for convolutional noise removal [2], it still entails a few simplifying assumptions. In this section we present two novel algorithms that remove these assumptions.

2.1 Spectral Equalization in the log DFT Domain

We first compare CMN to a different approach for the removal of stationary convolutional noise, “log-DFT mean normalization” (LDMN), and show that CMN is suboptimal when the cepstrum is computed as a linear transformation of the filterbank log energies. Specifically, we show that CMN can remove stationary convolutional noise only when the magnitude of the DFT of the channel’s impulse response is constant in each spectral band of the filterbank. We also show that we can overcome this assumption by equalizing the spectrum in the log-DFT domain.

In a filterbank-based front end, the DFT energies are integrated to compute the mel-filterbank energies. The log filterbank energies are used to compute the mel-cepstrum, which is normalized by removing its mean in each sentence.

1 INTRODUCTION

In many practical situations, an automatic speech recognizer has to operate in various but well-defined acoustic environments. The training corpus, however, is usually recorded with acoustic conditions that may not exactly match those encountered in the field. This mismatch between the acoustics of the training and testing data will degrade the accuracy of the recognizer. To overcome the data mismatch problem without collecting a new training corpus for each acoustic environment, we need a representation of the speech signal that is invariant across the acoustic spaces. Our purpose is to evaluate different techniques that facilitate the construction of acoustic models for speech recognition applications over a telephone channel.

The traditional approach to building speech recognizers is to collect training data under conditions that match as closely as possible the environment in which the recognizer will be used. To attain the best possible recognition performance, researchers typically try to match the language characteristics and acoustic environment in the training and testing phases. However, if there is no mismatch between the language characteristics of the training and testing data, then one can alternatively use algorithms to correct the *acoustic* mismatch between the training and testing corpora. This approach eliminates the need to collect speech data for each new acoustic environment. We will follow a twofold algorithmic approach to the acoustic mismatch problem. We first use a channel equalization algorithm that minimizes the channel mismatch between training and testing. We will compare a number of different equalization algorithms that remove some of the simplifying assumptions in the widely used sentence-based cepstral-mean removal, and show that the simple cepstral-mean removal algorithm is highly effective in correcting channel distortions. Once the channel distortion is reduced, our second main goal is to select a front end that is suitable for the testing conditions. In telephone applications, for example, the spectral bandwidth of the channel is limited to 3 kHz. Most of the spectral energy and the relevant information required for speech recognition are contained in this range. Hence, limiting the bandwidth can only increase the robustness of the recognizer to channel distortions and background noise. We show that by designing an appropriate front end for

Training Issues and Channel Equalization Techniques for the Construction of Telephone Acoustic Models Using a High-Quality Speech Corpus[1]

L. Neumeyer
415-859-4522
leo@speech.sri.com

V. Digalakis
415-859-5540
vas@speech.sri.com

M. Weintraub
415-859-6129
mw@speech.sri.com

SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
Fax: 415-859-5984

November 15, 1993

EDICS SA 1.6.4 and SA 1.6.8

ABSTRACT

We describe an approach for the estimation of acoustic phonetic models that will be used in a hidden Markov model (HMM) recognizer operating over the telephone. We explore two complementary techniques to developing telephone acoustic models. The first technique presents two new channel compensation algorithms. Experimental results on the *Wall Street Journal* corpus show no significant improvement over sentence-based cepstral-mean removal. The second technique uses an existing “high-quality” speech corpus to train acoustic models that are appropriate for the *Switchboard Credit Card* task over long-distance telephone lines. Experimental results show that cross-database acoustic training yields performance similar to that of conventional task-dependent acoustic training.

[1] This research was supported by the Advanced Research Projects Agency under Contract ONR N00014-93-C-0142 and ONR N00014-92-C-0154. It was also supported by a Grant, NSF IRI-9014829, from the National Science Foundation, and by SRI International internal research and development funds. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Project Agency or the National Science Foundation.