Learning-Based Auditory Encoding for Robust Speech Recognition

Yu-Hsiang Bosco Chiu, *Student Member, IEEE*, Bhiksha Raj, *Member, IEEE*, and Richard M. Stern, *Member, IEEE*

***Abstract*—This paper describes an approach to the optimization of the nonlinear component of a physiologically motivated feature extraction system for automatic speech recognition. Most computational models of the peripheral auditory system include a sigmoidal nonlinear function that relates the log of signal intensity to output level, which we represent by a set of frequency dependent logistic functions. The parameters of these rate-level functions are estimated to maximize the *a posteriori* probability of the correct class in training data. The performance of this approach was verified by the results of a series of experiments conducted with the CMU S phinx-III speech recognition system on the DARPA Resource Management, Wall Street Journal databases, and on the AURORA 2 database. In general, it was shown that feature extraction that incorporates the learned rate-nonlinearity, combined with a complementary loudness compensation function, results in better recognition accuracy in the presence of background noise than traditional MFCC feature extraction without the optimized nonlinearity when the system is trained on clean speech and tested in noise. We also describe the use of lattice structure that constraints the training process, enabling training with much more complicated acoustic models.**

***Index Terms*—Auditory model, discriminative training, feature extraction, robust automatic speech recognition.**

I. Introduction

THE human auditory system serves a wide range of functions our daily life, enabling the encoding and recognition of a diversity of environmental sounds such as human speech, animal songs, and background noises. An essential component of this task is the accurate representation of the relative intensity of an incoming sound as a function of frequency. While the method by which the auditory system encodes the intensity of sound is still under debate [1]–[4], one can argue that it is likely to be optimized at some level for the recognition of human speech sounds.

It is often hypothesized that the various aspects of human auditory perception, such as frequency resolution of the cochlea [5], [6], nonlinear compressive effects of the middle ear [7], [8], simultaneous and non-simultaneous masking effects [9]–[12], etc., aid or enhance human ability to recognize speech, particularly in the presence of noise. Researchers have therefore attempted to incorporate many of these attributes into the feature extraction stages of automatic speech recognition systems as well with varying degrees of success (e.g., [13], [14]).

Prior attempts at modeling the human auditory system may broadly be divided into two categories—those that attempt to mimic various aspects of the auditory system, usually through empirically or mathematically derived analytical models of auditory processes (e.g., [15] and [16]), and those that only retain the *framework* of auditory processing, but actually optimize model parameters for automatic speech recognition (e.g., [13], [14]).

The latter approach is particularly attractive for the following reason: it is reasonable to believe that biological auditory processes have been optimized for the manner in which the brain processes and recognizes sounds (subject to other physiological constraints). It is questionable that the detailed structure of human auditory processing is also optimal for automatic speech recognition systems, which are complex statistical machines whose relationship to the actual recognition processes in the brain is unknown. It follows that if we were to optimize the parameters of auditory processing for automatic speech recognition, the resultant feature computation module is likely to result in superior performance compared to features obtained by blind mimicry of auditory models.

Most prior attempts at optimizing the parameters of a physiologically motivated feature computation scheme for automatic recognition have concentrated on the filter bank that is used for frequency analysis. For example, Biem *et al.* propose a discriminative feature extraction procedure which refines the filter bank, by using a smoothed binary loss [17]. Kinnunen used the F-ratio to design a filter bank for improving speaker recognition performance [18]. These methods have primarily addressed data-driven optimization of the frequency analysis of the speech signal. Other authors have attempted to modify the nonlinear compression of feature computation for better speech processing [19] and recognition, e.g., [14], [20]–[23]. Chatterjee *et al.* proposed an augmentation of MFCC features by including higher-order terms of filter bank energy outputs and optimizing them such that the features extracted were similar in terms of the local geometries to the output of auditory model

Manuscript received December 31, 2010; revised April 19, 2011; accepted September 04, 2011. Date of publication September 15, 2011; date of current version nulldate. This work was supported in part by the National Science Foundation under Grants IIS-0420866 and IIS-I0916918, in part by DARPA, and in part by the Charles Stark Draper Laboratory University Research and Development Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark J. F. Gales.

Y.-H. B. Chiu is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15232 USA (e-mail: ychiu@cs.cmu.edu).

B. Raj is with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15232 USA (e-mail: bhiksha@cs.cmu.edu).

R. M. Stern is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15232 USA. He is also with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15232 USA (e-mail: rms@cs.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2011.2168209

[22], [23]. However, their objective is to develop a model that most closely mimics the outputs of actual auditory processing, without particular regard to automatic speech recognition performance.

In this paper, we investigate a technique for the design of yet another physiologically motivated processing stage in feature computation that is optimized for recognition accuracy. In previous work [24], we have determined that the *rate-level nonlinearity* that models the nonlinear relationship between input signal level and the putative rate of firing of fibers of the auditory nerve is a major contributor to robustness in speech recognition. In other physiological studies in cats it has been observed that the distribution of different types of auditory neurons with respect to spontaneous rate of activity depends on the amount of noise in the environment in which the animal was raised [25], indicating that the auditory-nerve response is at least partially a function of the "training" data to which the animal had been exposed. Motivated by these facts, we investigate a technique for automatically learning the parameters of a nonlinear compressive function that mimics the rate-level nonlinearity to optimize recognition accuracy in noise.

We show that we are able to learn a nonlinearity that does indeed improve recognition accuracy significantly in the presence of noise. Additionally, we show that the performance of the learned rate-level nonlinearity has both generalizable and task-specific aspects, validating our hypothesis that the parameters must be *learned* since their optimal values may be different for different tasks.

The rest of this paper is organized as follows. In Section II, we describe the feature computation scheme we will employ, that incorporates a stage modeling the rate-level nonlinearity. In Section III, we describe the learning algorithm to learn the parameters of the nonlinearity. Automatic learning of the parameters can be a computationally expensive process. We also discuss in Section III two ways to reduce the computational complexity associated with the learning process: the use of conjugate gradient descent to reduce the total number of iterations for achieving convergence, and restriction of the gradient search to legal candidate states according to a lattice of allowable word sequences in the training data. In Section IV, we describe experiments conducted on the DARPA Resource Management, Wall Street Journal and AURORA 2 corpora in the presence of several types of noise. Finally, our summary and conclusions are provided in Section V.

II. Feature Computation Using a Rate-Level Nonlinearity and Equal-Loudness Compensation

Most physiologically motivated feature extraction schemes take the form of concatenation of a bank of bandpass filters, a rectifying nonlinearity, and a subsequent additional filtering and other processing components that vary from implementation to implementation (e.g., [15], [16], [26]).

A significant aspect of the human auditory system is a nonlinear relationship between the loudness of perceived sound and neuronal firing rate. Nearly all physiologically motivated feature extraction schemes model this relationship. Typically this is done by a logarithmic or power-law nonlinearity. In the Seneff model in particular [16], this is modeled by a *rate-level* nonlinearity, which operates as a soft clipping mechanism that limits the response to both very small and very large amplitudes of sound.

In a previous study [24], in which we analyzed the contributions of various elements of the Seneff model to speech recognition performance, we determined that the rate-level nonlinearity is the element that provides the greatest robustness with respect to additive noise. The rate-level nonlinearity in auditory models differs from the usual power-law and logarithmic compression used in root-power or mel-frequency cepstra, in that it not only compresses high signal levels, but also low ones. A typical nonlinearity, as abstracted from a model of the peripheral auditory system [27], is shown in the solid curve in the upper left panel of Fig. 1; the dashed curve depicts the traditional logarithmic rate-level nonlinearity used in MFCC and similar processing. Small-amplitude sounds are more easily affected by noise. By nonlinearly compressing small-amplitude signals, the rate-level nonlinearity appears to reduce the effects of noise, resulting in reduced degradation of recognition accuracy.

The lower left panels of Fig. 1 depict separately the amplitude histograms of clean speech in training data, and white noise, with a signal-to-noise ratio (SNR) of 20 dB. Note in these panels that the responses to the speech component are in the graded part of the rate-intensity function while the responses to the less-intense noise fall in the portion of the rate-intensity curve for which the output remains relatively constant independently of the input. In the right panels of the same figure, we show the spectra derived after the traditional log compression (upper right panel) and using the physiologically derived rate-level function (lower left panel). In each case, responses are shown for clean speech and speech degraded by white noise at an SNR of 20 dB, corresponding to the solid and dashed curves, respectively. As can be seen in the figure, the use of the nonlinear rate-intensity function sharply reduces the differences between the shapes of the curves representing clean speech from speech in noise.

As noted above, we argued in [24] that the most important aspect of the auditory model was the nonlinearity associated with the hair cell model. To the extent that this is true, we should be able to obtain a similar benefit by applying such a nonlinearity to conventional MFCC-like feature extraction. Toward this end we modeled the nonlinear curve in the upper left panel of Fig. 1 by a logistic function and interposed it between the log of the triangularly weighted frequency response and the subsequent discrete Fourier transform (DCT) operation in traditional Mel-frequency cepstral coefficient (MFCC) processing, as shown in Fig. 2 ([24], [28]–[30]). Specifically, after windowing the incoming signal into frames of brief duration, a short-time Fourier Transform is applied to obtain the power spectrum of each frame. The power spectrum is then integrated into a Mel-spectrum using traditional triangle-shaped weighting functions to obtain the equivalent of the output of a Mel-frequency filterbank. The filterbank output is then compressed by a logarithmic nonlinearity.

An additional aspect of psychoacoustic models, which we also evaluated as part of the feature computation in [24], is an *equal-loudness weighting* shown in Fig. 3 that is derived

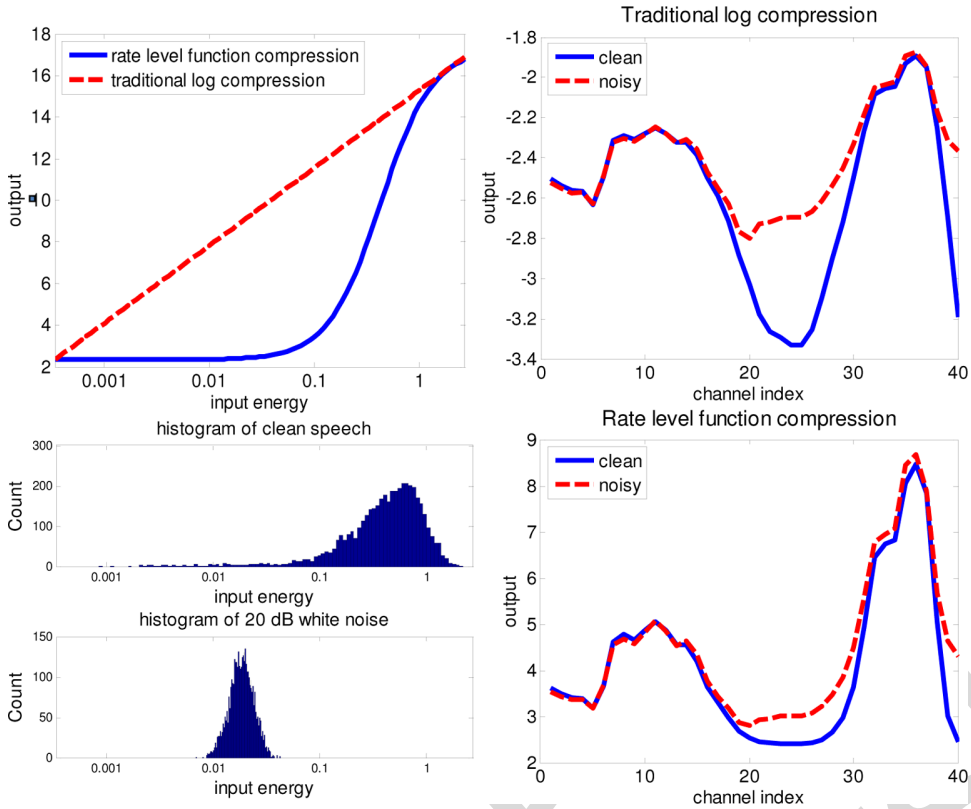


Fig. 1. Upper left panel: physiologically motivated rate-level function in the half wave rectification stage (solid curve) compared with traditional log compression (dashed line). Lower left panels: magnitude (rms) histogram for clean speech and for white noise, with an SNR of 20 dB. Right panels: log Mel spectrum under clean conditions (solid line) and in white noise at an SNR of 20 dB (dashed line). Responses are compared for traditional logarithmic compression and for the rate-level function discussed in this paper (upper and lower right panels, respectively).

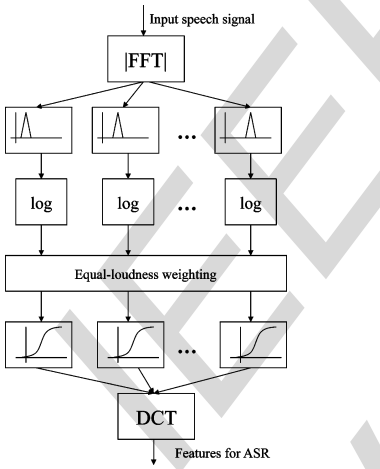


Fig. 2. Block diagram of the feature computation scheme. Note that a frequency weighting and a sigmodial nonlinearity are interposed between the log transformation and the DCT in traditional MFCC processing.

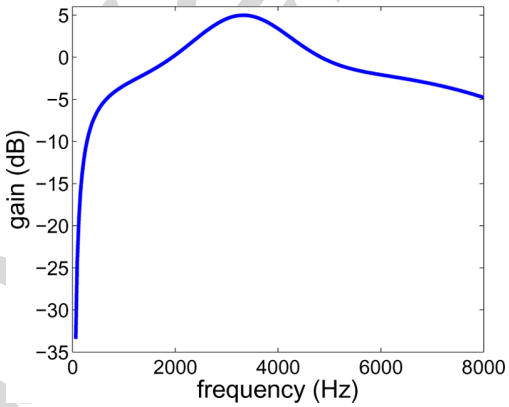


Fig. 3. Function used to approximate equal-loudness weighting based on the results in [31].

from the equal-loudness curve [31] which characterizes psychoacoustical results relating signal intensity to perceived loudness. While in reality perceived loudness depends on both the frequency and intensity of the incoming signal, we only normalize the mean response and assume that it is dependent only on frequency.

In computational models, equal-loudness weighting is implemented as a constant, frequency-dependent multiplicative weighting of the filter-bank output. In our implementation we apply it instead as an additive correction to the logarithm-compressed mel-frequency filterbank output, which is why the equal-loudness weighting appears after the log operation in Fig. 2.

The equal-loudness weighted log-mel-spectrum is then passed through a logistic function that is introduced to model the nonlinear average auditory-nerve response as a function of the input level in decibels

$$x_{u,t}[n] = \frac{\alpha[n]}{1 + exp(w_1[n] \cdot y_{u,t}[n] + w_0[n])} \qquad (1)$$

where $y_{u,t}[n]$ is the nth log Mel-spectral value and $x_{u,t}[n]$ is the corresponding sigmoid-compressed value of frame t in utterance u. The parameters of the nonlinearity, $\alpha[n] = 0.05$; $w_0[n] = 0.613$; $w_1[n] = -0.521$, $\forall\, n$, were determined empirically by evaluation on the Resource Management development with white noise added at an SNR of 10 dB. These parameter values are used in all our experiments. Note that these values are the same for all Mel-frequency components, i.e., they are frequency independent. Finally, cepstral-like coefficients were obtained by applying the DCT transform to the output of the rate-level nonlinearity.

A final note on equal-loudness weighting: in conventional feature computation that employs a logarithmic nonlinearity, the equal-loudness weighting is canceled out by the cepstral mean subtraction (CMS) that is routinely used in speech recognition. This is one reason why it is generally not used in Mel-frequency cepstral computation, but remains a part of some other feature computation schemes such as PLP, which use other forms of compression. In our model too, logistic compression following the logarithmic compression ensures that the equal-loudness weighting is not canceled out by CMS.

III. Learning the Nonlinearity

The premise of our paper is that the nonlinearities in the human auditory tract are a function of more than mere recognition performance (a view endorsed by Bregman [32] among others), but nevertheless serves a demonstrably useful purpose in recognition. In a computational model that need not consider other factors not related to recognition, the principle behind the nonlinearity could be retained, while the actual form with which it is implemented could be explicitly optimized for recognition performance.

Rather than hypothesize an entirely new form for the nonlinearity however, we retain the sigmoidal form described in (1), but attempt to determine the *parameters* of the nonlinearity to optimize recognition accuracy obtained with an automatic speech recognition system.

Unfortunately, the hidden Markov models for the various phonemes and the language model used for automatic speech recognition are quite complex, and it is difficult to obtain a simple update mechanism that can relate recognition accuracy to the parameters of the sigmoidal nonlinearity. Because of this, we use a simple Bayesian classifier for sound classes in the language as a substitute for the recognizer itself. Each sound class is modeled by a Gaussian distribution, computed from training data for that sound class. We use a maximum-mutual information (MMI) criterion to estimate the parameters of the nonlinearity such that the posterior probabilities of the phonemes based on their own training data are maximized.

The basic formulation for MMI training [33], [34] is well known. Given a parametric model $P(\mathbf{s}, C; \theta)$ expressing the joint probability distribution of data $\mathbf{s}$ and a class label C with parameters θ, MMI training learns θ such that the mutual information $I_\theta(\mathbf{s}, C)$ is maximized. The mutual information between $\mathbf{s}$ and C is given by

$$I_\theta(\mathbf{s}, C) = \log(\frac{P(\mathbf{s}, C)}{P(C)P(\mathbf{s})}) = \log P(C|\mathbf{s}) - \log P(C). \quad (2)$$

In the above equation, we have not explicitly represented θ, with the understanding that it represents the set of parameters of the model. Thus, for a given $P(C)$, maximizing the mutual information is equivalent to maximizing $\log P(C|\mathbf{s})$, where *a posteriori* probability is given by the usual Bayesian decomposition

$$P(C|\mathbf{s}) = \frac{P(\mathbf{s}|C)P(C)}{\sum_{C'} P(\mathbf{s}|C')P(C')} = \frac{P(\mathbf{s}|C)}{\sum_{C'} P(\mathbf{s}|C')} \quad (3)$$

with equal prior probabilities assigned to all sound classes. In our problem, C represents the sound classes, $\mathbf{s}$ represents the set of sequences of feature vectors for the recordings in our training set, i.e., $\mathbf{s} = \{\mathbf{s}_{u_1}, \mathbf{s}_{u_2}, \ldots\}$, where $\mathbf{s}_u$ is the sequence of feature vectors for any utterance u in our training set, and $\mathbf{s}_u = [\mathbf{s}_{u,1}, \mathbf{s}_{u,2}, \ldots]$ where $\mathbf{s}_{u,t}$ is the tth feature vector in $\mathbf{s}_u$.

We will also make use of the following approximations. We assume that individual utterances u are mutually statistically independent. We also assume that the *a posteriori* probability of the *true* label for an utterance, $C_u = \{C_{u,t},\ t = 1 \cdots T\}$ (individual vectors in the utterance may have different labels) is the product of the *a posteriori* probability of the individual vectors

$$\begin{aligned} P(C_u|\mathbf{s}_u) &= P(C_u)\frac{P(\mathbf{s}_u|C_u)}{\sum_{C'_u} P(\mathbf{s}_u|C'_u)P(C'_u)} \\ &\approx P(C_u)\prod_{t=1}^{T}\frac{P(\mathbf{s}_{u,t}|C_{u,t})}{\sum_{C'_{u,t}} P(\mathbf{s}_{u,t}|C'_{u,t})P(C'_{u,t})}. \end{aligned} \quad (4)$$

In other words, we assume that $\log P(C_u|\mathbf{s}_u) = \sum_{t=1}^{T} \log P(C_{u,t}|\mathbf{s}_{u,t})) + \log P(C_u)$. This approximation, which actually occurs in the denominator of the last term in the equation (which must ideally be summed over all class and HMM-state sequences), ignores the dependencies between class labels of adjacent vectors. It also ignores the contributions of the transition probabilities of the HMMs. These approximations greatly enhance the tractability of the problem, as explicitly incorporating these dependencies would greatly complicate the optimization. We also observed in pilot studies that including the transition probabilities into the estimation did not enhance the performance of the algorithm.

The actual optimization is performed using gradient descent. This is illustrated by Fig. 4.

The procedure for optimizing the nonlinearity is as follows. We assume we have a collection of training recordings: $\mathbf{s} = \{\mathbf{s}_{u_1}, \mathbf{s}_{u_2}, \ldots\}$ and their true labels $C = \{C_{u_1}, C_{u_2}, \ldots\}$.

Let $\boldsymbol{\mu}_C$ be the mean vector and $\boldsymbol{\sigma}_C$ be the covariance of the feature vectors for any sound class C. The likelihood of any vector $\mathbf{s}_{u,t}$, as computed by the distribution for that sound class is assumed to be given by a Gaussian density $N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})$. Further, we also assume that the individual classes $C_{u,t}$ are equally likely. This assumption not only simplifies our computation; in practice it was not observed to affect our results. The posterior probability of any sound class $C_{u,t}$, given a specific observation $\mathbf{s}_{u,t}$ is given by

$$P(C_{u,t}|\mathbf{s}_{u,t}) = \frac{N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})}{\sum_{C'} N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})} \quad (5)$$

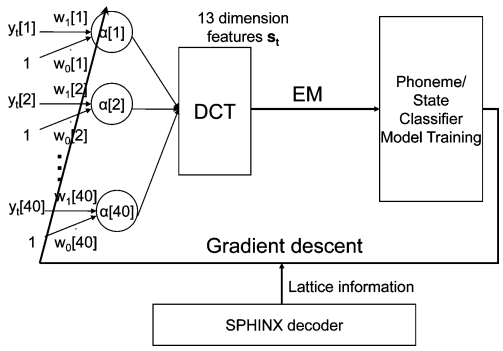


Fig. 4. Integrated system that refines the parameters characterizing the rate-level nonlinearity using the accuracy of a simple phonetic classifier as the objective function.

under the assumption that the prior probabilities of each class are equal.

The total overall log *a posteriori* probability of the true labels C of $\mathbf{s}$ is given by

$$\log P(C|\mathbf{s}) = \log P(C) + \sum_u \sum_t \log \frac{N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})}{\sum_{C'} N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})} \tag{6}$$

where u sums over all utterances and t sums over all features vectors for each utterance. Thus, optimizing $\log P(C|\mathbf{s})$ is equivalent to optimizing $\sum_u \sum_t \log(N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})/\sum_{C'} N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'}))$.

Our objective is to estimate the parameters of the sigmoidal nonlinearity of (1) to optimize $\log P(C|\mathbf{s})$ (for brevity, we will simply refer to $\log P(C|\mathbf{s})$ as $\log(P)$ henceforth). In doing so, however, it must also consider other aspects of the computation. Cepstral mean subtraction is a common component of speech recognition systems and is employed by us. The optimization algorithm must take this into consideration. In other words, we will actually optimize

$$\log(P) = \sum_u \sum_t \log \frac{N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})}{\sum_{C'} N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})} \tag{7}$$

where $\mathbf{s}^c_{u,t}$ is the *mean normalized* feature vector: $\mathbf{s}^c_{u,t} = \mathbf{s}_{u,t} - T_u^{-1} \sum_t \mathbf{s}_{u,t}$, where T_u is the number of features vectors in the utterance u. Here we have also ignored $\log P(C)$ as being irrelevant to our algorithm.

Also, modifying the manner in which features are computed will also modify the Gaussian distributions of the classes. Hence, the parameters of the Gaussian distributions of each sound class, and those of the sigmoidal nonlinearity in the feature computation, are jointly estimated to maximize $\log(P)$.

A. Estimating the Sound-Class Distribution Parameters

The model parameters $\boldsymbol{\mu}_C$ and $\boldsymbol{\sigma}_C$ for each sound class are initialized by training HMMs for all sound units using the conventional Baum–Welch algorithm, and conventional MFCC features. Thereafter, they are updated using the same objective criterion employed by the speech recognizer. For maximum-likelihood training, this is given by

$$\begin{aligned}
\boldsymbol{\mu}_C &= \frac{1}{\sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C)} \sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C) \mathbf{s}^c_{u,t}, \\
\boldsymbol{\sigma}_C &= \frac{1}{\sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C)} \\
&\quad \cdot \sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C) \left(\mathbf{s}^c_{u,t} - \boldsymbol{\mu}_C\right) \left(\mathbf{s}^c_{u,t} - \boldsymbol{\mu}_C\right)^T
\end{aligned} \tag{8}$$

where $I(\mathbf{s}^c_{u,t} \in C)$ is an indicator function that takes a value of 1 if $\mathbf{s}^c_{u,t}$ belongs to sound class C and 0 otherwise.

B. Estimating the Parameters of the Sigmoidal Nonlinearity

The parameters for the logistic function $\mathbf{F} = \{\boldsymbol{\alpha}, w_0, w_1\}$ are estimated by maximizing $\log(P)$ using a gradient descent approach. Taking the derivative of the objective function with respect to $\mathbf{F}$, the nonlinear parameters are updated as

$$\begin{aligned}
\boldsymbol{\alpha}^{\text{new}} &= \boldsymbol{\alpha}^{\text{old}} + 0.001 \cdot \gamma \cdot \frac{\partial \log P}{\partial \boldsymbol{\alpha}} \\
\boldsymbol{w}_0^{\text{new}} &= \boldsymbol{w}_0^{\text{old}} + \gamma \cdot \frac{\partial \log P}{\partial \boldsymbol{w}_0} \\
\boldsymbol{w}_1^{\text{new}} &= \boldsymbol{w}_1^{\text{old}} + 0.2 \cdot \gamma \cdot \frac{\partial \log P}{\partial \boldsymbol{w}_1}.
\end{aligned} \tag{9}$$

The forms of the partial derivatives are provided in Appendix. The weighting terms 0.001 and 0.2 were empirically obtained factors intended to result in roughly equal convergence rates for all three parameters, and the step size γ is equal to 0.05 in our experiments.

Our objective is to derive sigmoidal parameters minimizing the distortion in the features that results from corruption by noise. Thus, while class distribution parameters are learned from clean data, the sigmoidal parameters are learned to optimize classification on both clean and noisy data.

Thus, the updates of (9) are performed on both clean and noisy data, whereas the model updates of (8) are performed on clean data. After each step of gradient descent according to (9), the model parameters are updated using (8) on the clean training set only.

The procedure is iterative. Finally, once the objective function $\log(P)$ has converged, the nonlinearity parameters $\mathbf{F} = \{\boldsymbol{\alpha}, \boldsymbol{w_0}, \boldsymbol{w_1}\}$ are retained for the feature extraction process.

The model parameters of the entire speech recognition system are then retrained using features derived using the learned nonlinearity from the clean training set.

The entire learning algorithm is described in Algorithm 1. Here $\mathbf{y}_{u,t}$ represents the set of log mel-spectra that are input to the sigmoidal nonlinearity in (1) and $\mathbf{F}$ represents the set of parameters for the sigmoidal nonlinearity, as mentioned earlier. The feature vector $\mathbf{s}_{u,t}$ is derived from $\mathbf{y}_{u,t}$ as

$$\mathbf{s}_{u,t} = DCT(sigmoid(\mathbf{y}_{u,t}, \mathbf{F})) \tag{10}$$

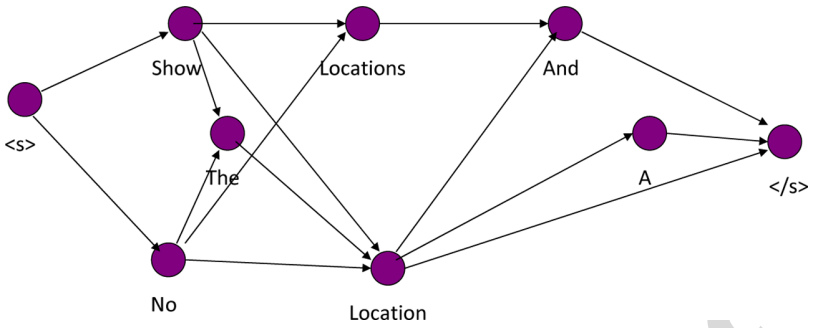


Fig. 5. Example of a word lattice to reduce the computational complexity by including only decoder-identified candidates as the competing classes.

as illustrated in Fig. 2, where $sigmoid(\mathbf{y}_{u,t}, \mathbf{F})$ represents the sigmoidal function of (1). Note that the sigmoid is applied individually to every spectral component in each log-mel-spectral vector y. Also, the sigmoidal parameters $\mathbf{F}$ are different for individual Mel-spectral channels. In addition, although Algorithm 1 only explicitly requires the features and does not require them to be mean normalized, the derivatives used in the update (given in the Appendix) are actually computed from mean-normalized features, thus explicitly accounting for CMS.

Algorithm 1: Algorithm for learning the parameters of the sigmoidal nonlinearity.

Input: $\mathbf{F}$, $\{\mathbf{y}_{u,t}, C_{u,t},\ u = 1 \ldots U, t = 1 \ldots T_u\}$

Output: $\mathbf{F}$

while *not converged* **do**

1. Compute the feature vector $\{\mathbf{s}_{u,t}\ \forall\ u,t\}$ from $\{\mathbf{y}_{u,t}\ \forall\ u,t\}$ using $\mathbf{F}$ in **(10)**.

2. Estimate $\{\boldsymbol{\mu}_C, \boldsymbol{\sigma}_C\}\ \forall\ C$ using **(8)** on the clean training set

3. Compute $\log(P)$ using **(7)** on both the clean and noisy training sets

4. $\mathbf{F}_{\text{new}} \leftarrow \mathbf{F}_{\text{old}} + \partial \log P / \partial \mathbf{F}$ using **(9)** on both clean and noisy training set

end

At convergence, the algorithm learns the optimal sigmoidal parameters $\mathbf{F}$ for each Mel-spectral channel.

C. Reducing Computational Complexity by Using a Word Lattice

A complete MMI solution that computes the ratio of the probability of the "true" class label to the sum of the probabilities of *all* classes can become prohibitively computationally expensive. The solution in the previous section assumes that each class is modeled by a single Gaussian. It is straightforward to extend it to include a mixture of M Gaussians (although, for pragmatic reasons we have not done so as we will explain in the concluding section of the paper). The amount of computation for calculating derivatives for each set of parameters at each iteration will be on the order of $O(KLMN)$, where K is the number of cepstral dimensions, L is the number of channels, and N is the number of sound classes. As the number of Gaussians increases with the complexity of the speech recognizer, the amount of computation becomes too large for feasible implementation.

The key reason for this computational explosion is the denominator in (7). For every observation, we must sum over all classes. To overcome this problem, we restrict the set of "competing" classes for each vector using a word lattice as shown in Fig. 5. Only the classes present in the word lattice are included in the MMI updates for any class: for each feature vector the set of competing classes that are considered when computing the *a posteriori* probability of the true class are only those classes that are present in the lattice at the same time instant as the vector. This affects the computation of both $\log P$ in (13) and the derivative of $logP$ in (14), in that the $\sum_{C'}$ becomes $\sum_{C'_{u,t} \in \text{activeclasses of } \mathbf{s}^c_{u,t} \text{ in the lattice}}$ in both equations. This results in a significant reduction of the number of competitors to be considered, and thereby the overall computation.

Using the word lattice in this manner also has a second effect. The lattice is obtained by recognizing the utterance using our initial acoustic models, along with a language model. The lattice hence represents the *a posteriori* most likely sequences of class labels, and thus implicitly factors in the distribution over class sequences into the objective function of (7), instead of simply marginalizing it out as (7) does.

In our experiments the word lattices were generated using the CMU Sphinx decoder on the training data using the initial acoustic model parameters. The lattices were saved and subsequently remained fixed throughout the optimization process.

D. Optimizing the Speed of Convergence Using Conjugate Gradient Descent

It is well known that the simple gradient-optimization approach, such as that followed in (9) tends to be slow: the gradients at consecutive iterations tend to have high correlation, as a result of which the steps taken at consecutive iterations are very similar and are somewhat redundant. The method of conjugate gradient descent [35], [36] avoids this problem by ensuring that the steps taken at consecutive iterations are orthogonal to one another in the parameter space. This dramatically increases the speed with which the solution is obtained.

We therefore modified our basic algorithm to implement the method of conjugate gradient descent. The modified algorithm is summarized in Algorithm 2. In the algorithm M is a $3L \times 3L$ diagonal weighting matrix, where L is the number of mel-frequency components in each mel-log-spectral vector. The diagonal entries of M are the weights used in (9)—the first L entries are 0.001, the next L are 1, and the final L diagonal entries are 0.2.

Algorithm 2: Algorithm for learning the parameters of the sigmoidal nonlinearity where $\sigma_0 = 0.05$ and $j_{max} = 5$. The matrix M is diagonal with entries equal to $[0.001, \ldots, 1, \ldots, 0.2, \ldots,]$, which are the weights used in **(9)**. The variable r represents the raw gradient, s is the scaled gradient, d is the conjugate gradient search direction, and β measures the projection of the previous search direction onto the current search direction. The inner loop performs j_{max} iterations of a line search in the search direction. The outer loop updates the search direction to a new orthogonalized gradient, or, if the projection is negative, to a new scaled gradient.

Input: $\mathbf{F}$, $\{\{\boldsymbol{y}_{u,t}, C_{u,t}\},\ u = 1 \ldots U, t = 1 \ldots T_u\}$

Output: $\mathbf{F}$

1. $r \leftarrow \partial \log P / \partial \mathbf{F}$ as developed in **Appendix**

2. $s \leftarrow Mr$ where M is a weighting matrix representing the weighting shown in **(9)**

3. $d \leftarrow s$

4. $\delta_{\text{new}} \leftarrow r^T d$

5. **while** *not converged* **do**

6. $j \leftarrow 0$

7. $\gamma \leftarrow \sigma_0$

8. **while** $j < j_{max}$ **do**

9. Compute feature vector $\{\mathbf{s}_{1,1}, \ldots, \mathbf{s}_{u,T_u}\}$ using **(10)**

10. Estimate $\{\boldsymbol{\mu}_C, \boldsymbol{\sigma}_C\}\ \forall\ C$ on the clean training set

11. Compute $\log(P)$ using **(7)** on both clean and noisy training set

12. $\eta \leftarrow [\partial \log P / \partial \mathbf{F}]^T d$

13. **if** $j \neq 0$ then

14. $\gamma \leftarrow \gamma(0.5\eta/\eta' - \eta)$

end

15. $\mathbf{F}_{\text{new}} \leftarrow \mathbf{F}_{\text{old}} + \gamma d$

16. $\eta' \leftarrow \eta$

17. $j \leftarrow j + 1$

end

18. $r \leftarrow \partial \log P / \partial \mathbf{F}$

19. $\delta_{\text{old}} \leftarrow \delta_{\text{new}}$

20. $\delta_{\text{mid}} \leftarrow r^T s$

21. $s \leftarrow Mr$

22. $\delta_{\text{new}} \leftarrow r^T s$

23. $\beta \leftarrow \delta_{\text{new}} - \delta_{\text{mid}} / \delta_{\text{old}}$

24. **if** $\beta \leq 0$ **then**

25. $d \leftarrow s$

else

26. $d \leftarrow s + \beta d$

end

end

IV. Experimental Results

Experiments were run on the DARPA Resource Management RM1 and the DARPA Wall Street Journal WSJ0 corpora to evaluate the methods that are proposed above. The Sphinx-III continuous-density HMM-based speech recognition system was used in all experiments. The feature extraction employed a 40-filter Mel filter bank covering the frequency range of 130 to 6800 Hz. Each utterance is normalized to have zero mean and unit variance before multiplication by a 25.6-ms Hamming window with frames updated every 10 ms.

A. Effect of Frequency Equalization

In our system implementation, each log spectral component is shifted by the equal loudness function shown in Fig. 3. As we mentioned before, this linear filtering does not affect the performance of traditional MFCC processing as it introduces an additive constant to the cepstral coefficients that is removed by cepstral mean subtraction (CMS). In contrast, the sigmoidal nonlinearity affects the frequency normalization in a nonlinear fashion and therefore it is not eliminated by CMS. To better understand the effect of gain in our feature extraction system, we compare system performance of our system with and without the frequency-normalization component.

The feature extraction scheme described in Fig. 2 was applied to utterances from the DARPA Resource Management RM1 database which consists of Naval queries. 1600 utterances from the database were used as our training set and 600 randomly selected utterances from the original 1600 testing utterances were used as our testing set. 72 speakers were used in the training set and another 40 speakers in the testing set, representing a variety of American dialects. We used CMU's SPHINX-III speech recognition system with 1000 tied states, a language model weight of 9.5 and phonetic models with eight Gaussian mixtures. Cepstral-like coefficients were obtained for the proposed system by computing the DCT of the outputs of the nonlinearity. The major difference between traditional MFCC processing and our present approach (both with and without the frequency weighting) is in the use of the rate-level nonlinearity described above. Cepstral mean subtraction (CMS) was applied, and delta and delta-delta cepstral coefficients were developed in all cases in the usual fashion. The parameters of the nonlinearity are $\alpha[n] = 0.05$; $w_0[n] = 0.613$; $w_1[n] = -0.521$, $\forall\ n$ as was mentioned in Section II.

Recognition experiments were run on speech corrupted by a variety of noises. The noises were obtained from the NOISEX-92 database (including a later release of NOISEX) [37], [38], and included recordings of speech babble, and real noise samples in a market, restaurant and theater. All of these noises are digitally added to the original clean test set at SNRs of 0, 5, 10, 15, and 20 dB. We plot recognition accuracy, which is computed as 100% minus the word error rate, where the latter is defined to be the ratio of the total number of insertion, deletion, and substitution errors divided by the number of incoming words.

Fig. 6 compares speech recognition accuracy of the proposed system with and without the equal loudness curve in the presence of four different types of background noise. The horizontal

axis represent the SNR of the test set and vertical axis represents recognition accuracy (calculated as 100%—WER). The filled squares and diamonds represent the recognition accuracy obtained using the rate-level nonlinearity with and without out equal loudness weighting, respectively, and the triangles and open triangles represent the same index using traditional MFCC processing. As can be seen from the figure, the equal loudness curve (which can be thought of as a manipulation of the parameter $w_0[n]$ in different frequency channels) substantially improves speech recognition accuracy, especially in natural environments such as the market or a theater when the rate-level nonlinearity is used. Frequency weighting has almost no impact on the performance of traditional MFCC processing, as expected. We will discuss the optimal parameter values in greater depth below.

B. Recognition Accuracy Using Optimized Nonlinear Parameters

Our sigmoidal rate-level nonlinearity is trained on clean speech from the RM1 database to which pink noise from the NOISEX-92 corpus was digitally added at an SNR of 10 dB. Class labels for training were based on an HMM with 1000 tied states that was generated by forced alignment of the clean training data using previously trained models. The noisy testing sets were created by artificially adding babble noise from the NOISEX-92 corpus, the recordings of market, theater and restaurant noises obtained in real environments to the original clean testing set. We note that the noises used in these training and testing environments were different. The step size was set to 0.05 to achieve stable but reasonably fast convergence.

The choice of 10-dB pink noise was based on preliminary experiments performed on a held-out data set [30]. In general, we found that as long as the energy distribution of the spectrum of the noise used for training is similar to that of the noise in the test data (e.g., the power spectrum decreases from low frequencies to high frequencies), the actual type of noise used for training does not matter. The actual SNR chosen is also supported by past experience: 10 dB tends to be close to the "knee" in plots of recognition error as a function of SNR. If the recognition performance at this noise level is improved, overall performance in the presence of noise tends to improve as well.

Fig. 7 shows the rate-level nonlinearities that were actually learned. Fig. 7(a) is a 3-D plot showing the nonlinearities for all 40 Mel-frequency channels. Fig. 7(b) depicts a few cross-sections of this plot. Fig. 7(c)–(e) show how the individual parameters of the rate-level nonlinearities vary as a function of frequency. We note that the estimated optimal rate-level functions vary greatly across frequencies in all aspects, including gain, slope, and attack.

In comparing the rate-level functions that are learned for different types of background noise, we have found that while the details of the resulting functions differ slightly, the general trends are similar, with a shallow slope in the middle to capture the large dynamic range of speech frequency components in the mid frequencies and a steeper slope in both the low- and high-frequency regions.

Once the parameters of the feature computation module were learned, the feature computation module was employed to derive features from a clean version of the RM training set, from which the HMM model parameters were retrained.

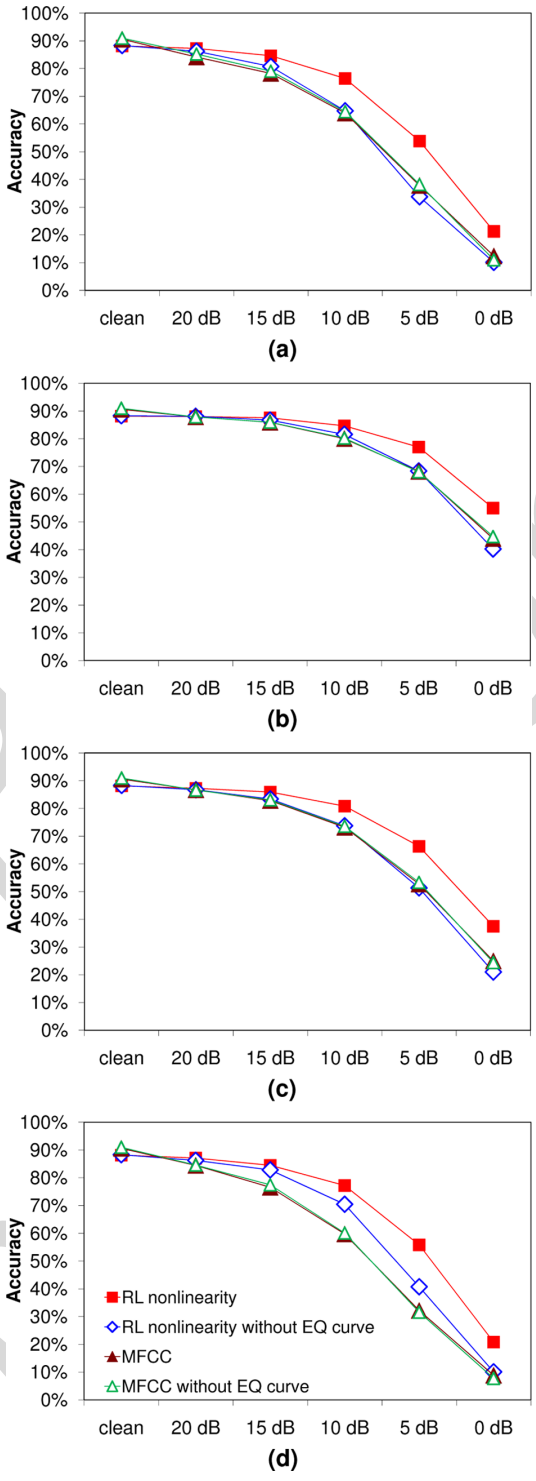


Fig. 6. Demonstration of the impact of the rate-level (RL) nonlinearity and the equal loudness curve. System recognition accuracy (100%—WER) is compared using the RL nonlinearity with equal loudness weighting (squares), the same system without equal loudness weighting (diamonds), baseline MFCC processing (triangles), and baseline MFCC processing without equal loudness weighting (empty triangles) for the RM database in the presence of four different types of background noise. The WERs obtained training and testing with clean speech are—RL nonlinearity: 11.88%, RL nonlinearity without weighting: 11.72% MFCC: 9.45%, MFCC without weighting: 9.07%. (a) Babble noise. (b) Market noise. (c) Restaurant noise. (d) Theater noise.

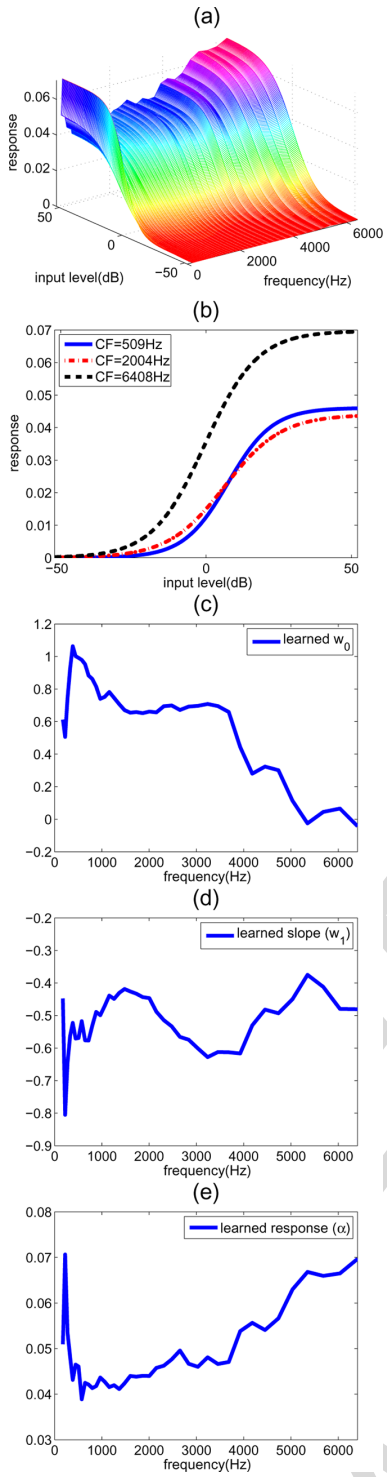


Fig. 7. (a) Trained RL nonlinearity as a function of input intensity and frequency. (b) Examples of the trained RL nonlinearity at selected low, mid and high frequencies. (c)–(e) The trained values of the logistic function parameters w_0, w_1, and α as a function of frequency.

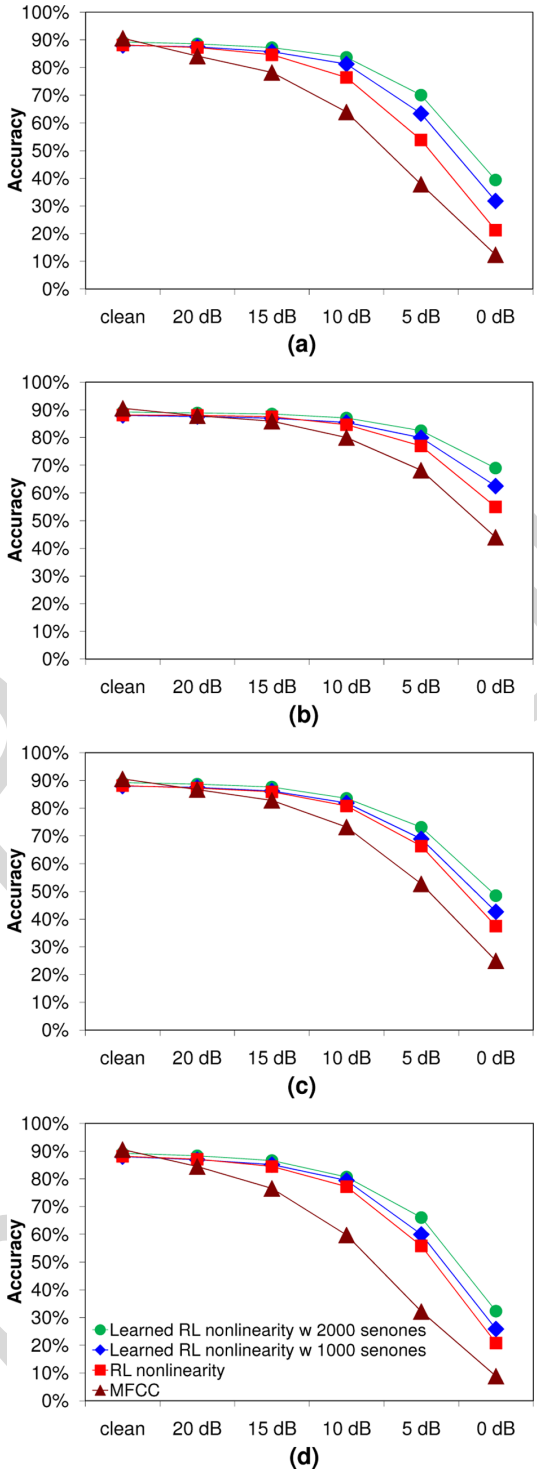


Fig. 8. Comparison of the recognition accuracy obtained using the optimized RL nonlinearity to that obtained with the baseline RL nonlinearity (without optimization) and with MFCC coefficients, all in the presence of four types of background noise using the RM corpus. The WERs obtained training and testing with clean speech are—MFCC: 9.45%, RL nonlinearity: 11.88% with p-value compared to MFCC: 0.00003, RL nonlinearity learned from 1000 tied states: 11.97%, p-value: 0.000013, RL nonlinearity learned from 2000 tied states: 10.53%, p-value: 0.055. (a) Babble noise. (b) Market noise. (c) Restaurant noise. (d) Theater noise.

Fig. 8 compares the recognition accuracy that was obtained using the optimized rate-level nonlinearity (training with 1000 and 2000 senones) to the corresponding results obtained using a similar nonlinearity derived from a model of physiological data [27] with some empirical tuning but no systematic optimization [30], and conventional MFCC coefficients. We note that

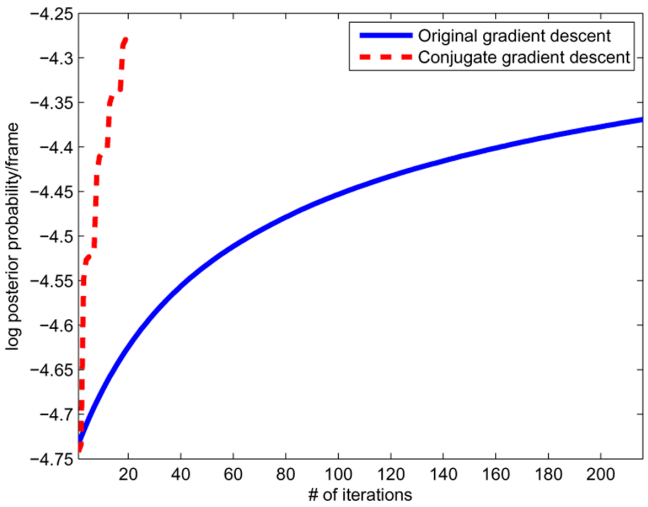


Fig. 9. Comparison of the log posterior probability as a function of the number of iterations using conjugate gradient descent (dashed curve) and traditional gradient descent (solid curve).

while the use of the frequency weighting and rate-level nonlinearity from physiological measurements greatly improves noise robustness compared to the MFCC baseline *(with accuracy loss under clean condition)*, the best results are obtained with the automatically learned parameters. As before, the noise types used for training and testing are different in these experiments.

C. Improvements to Training Speed

Improvement in training speed is provided by two major factors: a reduction in the time required for each iteration of the training procedure (through the use of the lattice representation described in Section III-C), and a reduction in the total number of iterations over the entire training process (which is provided by the use of conjugate gradient descent, as described in Section III-D).

The use of the lattice structure as described in Section III-C reduces the processing time per iteration by reducing the number of competing candidate hypotheses that need to be considered. In empirical comparisons of the processing time with and without the lattice representation we observed that the use of the word lattice reduces the processing time for each iteration of the gradient descent from an average of 727 to 291 s, a reduction factor of approximately 2.5.

Fig. 9 compares the *a posteriori* probability as a function of the number of iterations needed to achieve convergence. The dashed line describes convergence using the conjugate gradient descent method while the solid line describes convergence using the traditional gradient descent method, using the Resource Management database. As can be seen from the figure, the use of conjugate gradient descent reduces the number of iterations required to achieve convergence by a factor of at least 10 compared to traditional gradient descent. We have observed empirically that the performance of the recognition system approximates its optimal value after about 20 iterations of training.

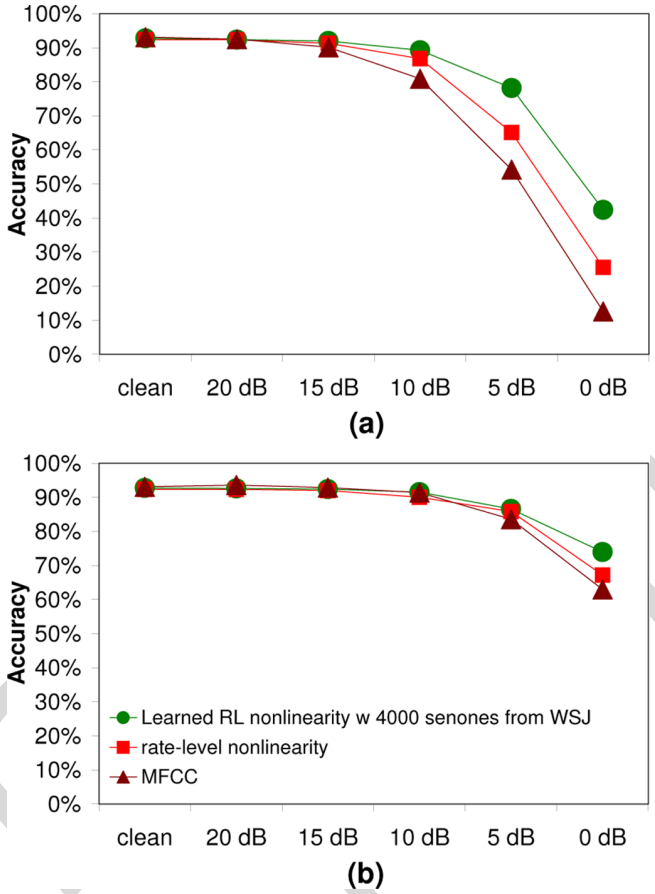


Fig. 10. Comparison of recognition accuracy in the presence of two types of background noise on the WSJ corpus, using procedures similar to those in Fig. 8. The WERs obtained training and testing with clean speech are—MFCC: 6.91%, RL nonlinearity: 7.66%, p-value compared to MFCC: 0.135, RL nonlinearity using 4000 tied states: 7.25%, p-value: 0.493. (a) Babble noise. (b) Market noise.

D. Recognition Accuracy Using the DARPA Wall Street Journal Database

We also evaluated our recognition system using the sigmoidal rate-level nonlinearity with optimized parameters on the standard DARPA Wall Street Journal (WSJ) database. The training set we used consisted of 7024 speaker-independent utterances from 84 speakers. The test set consisted of 330 speaker-independent utterances from the evaluation set of the 5000-word WSJ0 database, using non-verbalized punctuation. As with the Resource Management database, a noisy test set was created by artificially adding babble noise from the NOISEX-92 database and market noise from recordings in real environments at pre-specified SNRs of 0, 5, 10, 15, and 20 dB. The noisy training set was created by adding 10-dB pink noise from the NOISEX-92 database to the original clean training set. The SPHINX-III trainer and decoder were implemented using 4000 tied states, a language model weight of 11.5, and 16 components in all GMMs, with no further attempt made to tune system parameters. Other conditions are the same as in the RM case.

Fig. 10 shows results using the WSJ database for conditions that are similar to those depicted in Fig. 8 except that only a subset of testing noises are examined and a greater number of

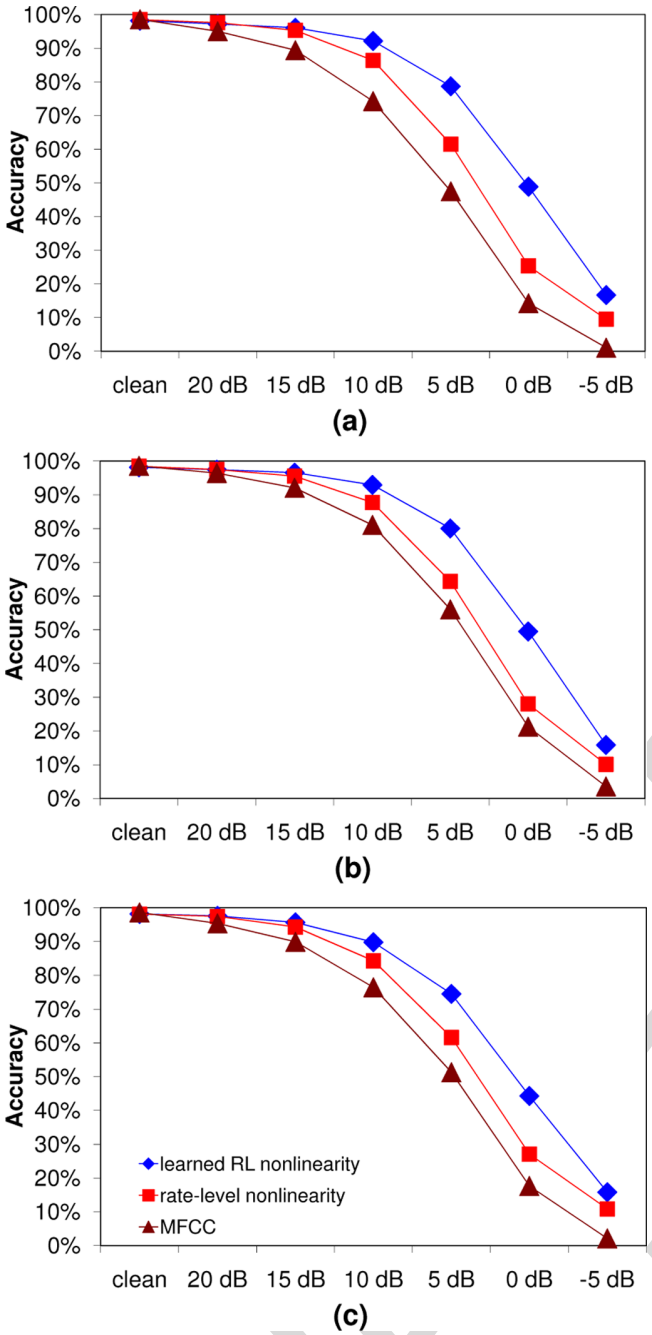


Fig. 11. Comparison of recognition accuracy in the presence of three sets of background noise from the AURORA 2 corpus. The WERs obtained training and testing with clean speech are MFCC: Test A 1.43%, Test B 1.43%, Test C 1.42%,RL nonlinearity: Test A 1.54%, p-value compared to MFCC: 0.461, Test B 1.54%, p-value: 0.461, Test C 1.93%, p-value: 0.023, learned RL nonlinearity: Test A 1.86%, p-value: 0.006, Test B 1.86%, p-value: 0.006, Test C 1.86%, p-value: 0.047. (a) Test set A. (b) Test set B. (c) Test set C.

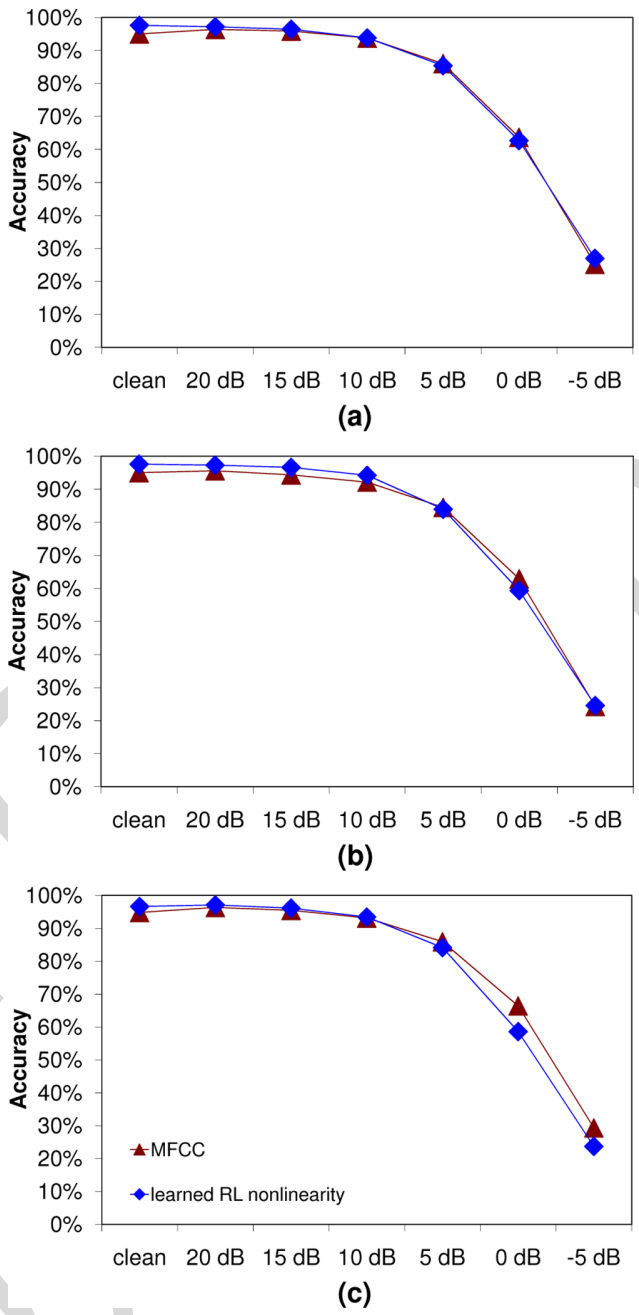


Fig. 12. Comparison of recognition accuracy in the presence of three sets of background noise for the AURORA 2 corpus using multi-style training. (a) Test set A. (b) test set B. (c) Test set C.

senones was used. These results confirm that recognition accuracy using the WSJ data follow trends that are similar to what has been previously described for the RM database. The optimization process provides an additional increase of 2 to 4 dB in effective SNR compared to the SNR obtained using the deterministic initial values of the parameters of the rate-level nonlinearity and an improvement of 3 to 5 dB compared to the baseline MFCC results.

E. Recognition Accuracy Using the AURORA 2 Database

Fig. 11 shows results obtained using the AURORA 2 database after training using clean speech. HMMs with 1000 tied states, each modeled by a mixture of eight Gaussians for MFCC coefficients, and 32 Gaussians for features obtained using the rate-level nonlinearity, were trained for recognition experiments. (It was found that the use of 32 Gaussians to characterize MFCC features provided better recognition accuracy for clean speech but higher error rates for the noisy conditions considered. 8 Gaussians per senone provided the best MFCC performance in the noisy evaluation conditions.) The feature extraction employed a 23-filter Mel filter bank covering the frequency range

of 64 Hz to 4000 Hz. The number of cepstral coefficients for best recognition accuracy was determined empirically to be 10 for MFCCs and 11 for the rate-level nonlinearity. The initial nonlinearity parameters were set to $w_0 = -0.110$, $w_1 = -0.521$, $\alpha = 0.05$ to account for the change of sampling rate from 16 kHz to 8 kHz.

The results of Fig. 11 indicate that recognition accuracy using the AURORA 2 database follows similar trends to what had been previously described for the RM and WSJ databases. The optimization process provides an additional 2 to 4 dB increase in effective SNR compared to the SNR obtained using the deterministic initials of rate-level nonlinearity and an improvement of 5 to 7 dB compared to the baseline MFCC results.

Fig. 12 shows results obtained using the AURORA 2 corpus with multi-condition training. The use of the recognition system with the learned sigmoidal rate-level nonlinearity does not appear to provide much benefit compared to baseline MFCC processing when multi-condition is employed.

V. Summary and Conclusion

In a previous study [24], we found that the sigmoidal rate-level nonlinearity that is a part of most models of the physiological transformation of sound to its neural representation contributes the most to robustness in speech recognition, especially when there is a mismatch of training and testing environments. In this paper we model this nonlinearity by a set of frequency-dependent logistic functions, and we develop an automated procedure for learning the optimal values of the parameters of these functions from training data using an objective function based on maximum mutual information. This function is coupled with a complementary function that models the observed psychoacoustical equal-loudness contour, and the two functions are inserted into the chain of operations that constitutes MFCC processing.

The process of learning the optimal parameters of the rate-level nonlinearities is sped up very substantially through the use of lattice information generated from the speech decoder to prune out unlikely state sequences, and through the use of conjugate gradient descent that reduces the total number of iterations required to achieve convergence. Together these improvements speed up the learning process by a factor of approximately 25.

Using equal-loudness compensation and the learned sigmoidal rate-level nonlinearity, we observed a typical improvement of approximately 5 to 7 dB in effective SNR compared to baseline MFCC processing at an SNR of 10 dB, and an improvement of 2 to 3 dB in effective SNR compared to a basic sigmoidal nonlinearity without the learning procedures described in this paper, when the system is trained on clean speech. These improvements in performance disappear when the system is trained and tested in multi-style fashion.

The algorithm described in Section III assumes that each of the phoneme classes is modeled by a single Gaussian. It is natural to hypothesize that having more detailed distributions, e.g., mixtures of Gaussians, could result in better learned sigmoidal parameters. The modifications required in the algorithm to deal with mixtures of Gaussians are minimal as only the *a posteriori* probabilities of individual Gaussians in the mixture need be considered. However, the increase in computation is significant, and in our experiments the benefit obtained from scaling up from single Gaussians to mixtures of Gaussians was marginal and did not justify the large increase in computation that it entailed.

Another natural question that arises is that of what happens if the sigmoidal parameters are learned from only clean speech. We note that the purpose of learning the parameters of the nonlinearity in the fashion described in this paper is to reduce the differences between features computed from clean speech and those obtained from noisy speech. Therefore, training sigmoidal parameters from noisy speech is an integral aspect of the algorithm. Nevertheless we did conduct an experiment where we learned the sigmoidal parameters from only clean speech. Not surprisingly, while performance on clean speech improved, it did not improve performance on noisy speech. This was to be expected: since the optimal nonlinearity is learned from data, it cannot possibly become robust to noise without being exposed to noise.

In a related study [30], we demonstrated that an additional improvement in recognition accuracy can be obtained by combining the learned rate-level nonlinearity with post-processing techniques such as modulation filtering of the cepstral-like coefficients that are derived from the processing described here. Nevertheless, we believe that further improvements can be obtained by fully integrating the benefits of all of these methods into a single algorithm that provides a joint optimization over both the parameters characterizing the nonlinearity parameters and the parameters that determine the modulation filter.

Appendix
Derivation of the Derivative Update Equation

With the assumption that the prior probabilities of each class are equal and that the observation probability $P(\mathbf{s}|C)$ is a single Gaussian, the feature vector $\mathbf{s}$ of the classifier can be computed from the input vector $\mathbf{x}$ using the rate-level nonlinearity and the DCT transformation

$$s_{u,t}[k] = \beta[k] \sum_{n=1}^{N} x_{u,t}[n] cos \frac{\pi(2n-1)(k-1)}{2N}$$
$$k = 1, \ldots, K, \; with$$
$$\beta[k] = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } k = 1 \\ \sqrt{\frac{2}{N}}, & \text{otherwise} \end{cases} \tag{11}$$

where $x_{u,t}[n]$ is given in (1) and K is the number of MFCC coefficients. (We used a value of $K = 13$ in the present paper.) The overall accumulated posterior probability can be written as

$$P = \prod_u \prod_t \frac{N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_C, \boldsymbol{\sigma}_C)}{\sum_{C'} N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})}. \tag{12}$$

In the above equations, u denotes the utterance index, t denotes the time index in each utterance, and $\mathbf{s}^c_{u,t}$ denotes the fea-

tures of the incoming utterance after cepstral mean substraction (CMS) has been applied, that $\mathbf{s}^c_{u,t} = \mathbf{s}_{u,t} - (1/T_u)\sum_{t=1}^{T_u}\mathbf{s}_{u,t}$:

$$\begin{aligned} &Objective = \max \log P \\ &= \max \sum_u \sum_t \left[-\frac{1}{2}\sum_{k=1}^{K}\left[\log\sigma_C[k]^2 + \frac{\|s^c_{u,t}[k]-\mu_C[k]\|^2}{\sigma_C[k]^2}\right] \right. \\ &\left. - \log \sum_{C'}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[k]^2}} e^{-(\|s^c_{u,t}[k]-\mu_{C'}[k]\|^2/2\sigma_{C'}[k]^2)}\right]. \end{aligned} \tag{13}$$

Taking the derivative with respect to $\mathbf{F} = \{\boldsymbol{\alpha}, w_0, w_1\}$ we obtain

$$\begin{aligned} &\frac{\partial \log P}{\partial \mathbf{F}} \\ &= \sum_u\sum_t\left[-\frac{1}{2}\sum_{k=1}^{K}\left[\frac{\partial \log\sigma_C[k]^2}{\partial\mathbf{F}} + \frac{\partial}{\partial\mathbf{F}}\frac{\|s^c_{u,t}[k]-\mu_C[k]\|^2}{\sigma_C[k]^2}\right]\right. \\ &\left. - \frac{\partial}{\partial\mathbf{F}}\log\sum_{C'}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[k]^2}} e^{-(\|s^c_{u,t}[k]-\mu_{C'}[k]\|^2/2\sigma_{C'}[k]^2)}\right] \end{aligned} \tag{14}$$

$$\begin{aligned} &\frac{\partial\{-\frac{1}{2}\log\sigma_C[k]^2\}}{\partial\mathbf{F}} = -\frac{1}{2\sigma_C[k]^2}\frac{\partial\sigma_C[k]^2}{\partial\mathbf{F}}, \\ &\frac{\partial}{\partial\mathbf{F}}\left[-\frac{1}{2}\frac{\|s^c_{u,t}[k]-\mu_C[k]\|^2}{\sigma_C[k]^2}\right] = \\ &-\frac{1}{2\sigma_C[k]^4}\left[2(s^c_{u,t}[k]-\mu_C[k])\left(\frac{\partial s^c_{u,t}[k]}{\partial\mathbf{F}} - \frac{\partial\mu_C[k]}{\partial\mathbf{F}}\right)\sigma_C[k]^2\right. \\ &\left. - \|s^c_{u,t}[k]-\mu_C[k]\|^2\frac{\partial\sigma_C[k]^2}{\partial\mathbf{F}}\right], \\ &\frac{\partial}{\partial\mathbf{F}}\left[\log\sum_{C'}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[k]^2}}e^{(-\|s^c_{u,t}[k]-\mu_{C'}[k]\|^2/2\sigma_{C'}[k]^2)}\right] \\ &= \frac{\frac{\partial}{\partial\mathbf{F}}\left[\sum_{C''}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C''}[k]^2}}e^{-(\|s^c_{u,t}[k]-\mu_{C''}[k]\|^2/2\sigma_{C''}[k]^2)}\right]}{\sum_{C'}\prod_{l=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[l]^2}}e^{-(\|s^c_{u,t}[l]-\mu_{C'}[l]\|^2/2\sigma_{C'}[l]^2)}} \\ &= \sum_{C''}\left[\sum_{k=1}^{K}\left[-\frac{1}{2\sigma_{C''}[k]^2}\frac{\partial\sigma_{C''}[k]^2}{\partial\mathbf{F}}\right.\right. \\ &\left.\left. - \frac{1}{2}\frac{\partial}{\partial\mathbf{F}}\left(\frac{\|s^c_{u,t}[k]-\mu_{C''}[k]\|^2}{\sigma_{C''}[k]^2}\right)\right]\right] \\ &\cdot\frac{\prod_{m=1}^{K}\frac{1}{\sigma_{C''}[m]}e^{-(\|s^c_{u,t}[m]-\mu_{C''}[m]\|^2/2\sigma_{C''}[m]^2)}}{\sum_{C'}\prod_{l=1}^{K}\frac{1}{\sigma_{C'}[l]}e^{-(\|sal^c_{u,t}[l]-\mu_{C'}[l]\|^2/2\sigma_{C'}[l]^2)}}. \end{aligned} \tag{15}$$

The model parameters $\boldsymbol{\mu}_C$ and $\boldsymbol{\sigma}_C$ were obtained in the maximum likelihood sense in the same fashion as in training the speech recognizer [(8) with mean substraction]:

$$\begin{aligned} \sigma_C[k] &= \frac{1}{\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C)}\sum_u\sum_t^{T_u} I(\mathbf{s}_{u,t}\in C) \\ &\cdot\left(s_{u,t}[k] - \frac{1}{T_u}\sum_{t=1}^{T_u}s_{u,t}[k] - \mu_C[k]\right)^2, \\ \mu_C[k] &= \frac{1}{\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C)}\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C) \\ &\cdot\left(s_{u,t}[k] - \frac{1}{T_u}\sum_{t=1}^{T_u}s_{u,t}[k]\right). \end{aligned} \tag{16}$$

The partial derivative of mean and variance of each class and feature vector $s^c_{u,t}[k]$ over $\mathbf{F}$ can be written as

$$\begin{aligned} \frac{\partial\sigma_C[k]^2}{\partial\mathbf{F}} &= \frac{2}{N_C}\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C) \\ &\times\left(\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} - \frac{1}{T_u}\sum_{t=1}^{T_u}\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} - \frac{\partial\mu_C[k]}{\partial\mathbf{F}}\right) \\ &\times\left(s_{u,t}[k] - \frac{1}{T_u}\sum_{t=1}^{T_u}s_{u,t}[k] - \mu_C[k]\right), \\ \frac{\partial\mu_C[k]}{\partial\mathbf{F}} &= \frac{1}{N_C}\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C)\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} \\ &- \frac{1}{T_u}\sum_u\sum_{t=1}^{T_u}\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}}, \\ \frac{\partial s^c_{u,t}[k]}{\partial\mathbf{F}} &= \frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} - \frac{1}{T_u}\sum_{t=1}^{T_u}\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} \end{aligned} \tag{17}$$

where N_C is the number of frames in class C and T_u is the number of frames in each utterance. We note that the above feature computation incorporates CMS. Furthermore, the mean of the sentence we are subtracting is not a constant developed during training, but rather it is taken as an average over the frames of the corresponding utterance. This occurs when we take the directive with respect to the nonlinearity parameters, which occurs in the second term of the last line of (17). In addition,

$$\begin{aligned} &\frac{\partial s_{u,t}[k]}{\partial\alpha[o]} = \frac{\partial}{\partial\alpha[o]} \\ &\cdot\left(\beta[k]\sum_{n=1}^{N}\frac{\alpha[n]}{1+e^{w_1[n]\cdot y_{u,t}[n]+w_0[n]}}cos\frac{\pi(2n-1)(k-1)}{2N}\right) \\ &= \beta[k]\frac{1}{1+e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}}cos\frac{\pi(2o-1)(k-1)}{2N} \\ &\frac{\partial s_{u,t}[k]}{\partial w_0[o]} \end{aligned}$$

$$= -\beta[k]\frac{e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}}{\left(1+e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}\right)^2}cos\frac{\pi(2o-1)(k-1)}{2N}$$

$$\frac{\partial s_{u,t}[k]}{\partial w_1[o]}$$

$$= -\beta[k]\frac{y_{u,t}[o]\cdot e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}}{\left(1+e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}\right)^2}cos\frac{\pi(2o-1)(k-1)}{2N}. \quad (18)$$

REFERENCES

[1] E. D. Young and M. B. Sachs, "Representation of steady state vowels in the temporal aspects of the discharge patterns of populations of auditory-nerve fibers," *J. Acoustic. Soc. Amer.*, vol. 66, pp. 1381–1403, 1979.

[2] N. F. Viemeister, "Auditory intensity discrimination at high frequencies in the presence of noise," *Science*, vol. 221, pp. 1206–1208, 1983.

[3] R. L. Winslow and M. B. Sachs, "Single tone intensity discrimination based on auditory-nerve rate responses in backgrounds of quiet, noise and stimulation of the crossed olivocochlcar bundle," *Hear. Res.*, vol. 35, pp. 165–190, 1988.

[4] I. M. Winter and A. R. Palmer, "Intensity coding in low-frequency auditory-nerve fibers of the guinea pig," *J. Acoust. Soc. Amer.*, vol. 90, no. 4, pp. 1958–1967, 1991.

[5] J. Volkmann, S. S. Stevens, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. Acoust. Soc. Amer.*, vol. 8, no. 3, pp. 208–209, 1937.

[6] E. Zwicker, "Subdivision of the audible frequency range into critical bands," *J. Acoust. Soc. Amer.*, vol. 33, no. 2, pp. 248–248, 1961.

[7] M. B. Sachs and P. J. Abbas, "Rate versus level functions for auditory-nerve fibers in cats: Tone-burst stimuli," *J. Acoust. Soc. Amer.*, vol. 56, pp. 1835–1847, 1974.

[8] M. B. Sachs, R. L. Winslow, and B. H. A. Sokolowski, "A computational model for rate-level functions from cat auditory-nerve fibers," *Hear. Res.*, vol. 41, pp. 61–70, 1989.

[9] M. B. Sachs and N. Y. Kiang, "Two-tone inhibition in auditory-nerve fibers," *J. Acoust. Soc. Amer.*, vol. 43, pp. 1120–1128, 1968.

[10] P. J. Abbas and M. B. Sachs, "Two-tone suppression in auditory-nerve fibers: Extension of a stimulus-response relationship," *J. Acoust. Soc. Amer.*, vol. 59, pp. 112–122, 1976.

[11] L. L. Elliott, "Changes in the simultaneous masked threshold of brief tones," *J. Acoust. Soc. Amer.*, vol. 38, pp. 738–746, 1965.

[12] E. Zwicker, "Temporal effects in simultaneous masking and loudness," *J. Acoust. Soc. Amer.*, vol. 38, pp. 132–141, 1965.

[13] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[14] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, pp. 1738–1752, 1990.

[15] O. Ghitza, "Auditory nerve representation as a front-end for speech recognition in a noisy environment," *Comput. Speech Lang.*, vol. 1, no. 2, pp. 109–131, 1986.

[16] S. Seneff, "A joint synchrony/mean-rate model of auditory speech processing," *J. Phonetics*, vol. 15, pp. 55–76, 1988.

[17] A. Biem, S. Katagiri, E. McDermott, and B.-H. Juang, "An application of discriminative feature extraction to filter-bank-based speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 9, no. 2, pp. 96–110, Feb. 2001.

[18] T. Kinnunen, "Design a speaker-discriminative adaptive filter bank for speaker recognition," in *Proc. Int Conf. Spoken Lang. Process.*, Denver, CO, Sep. 2002.

[19] T. Kobayashi and S. Imai, "Spectral analysis using generalized cepstrum," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 5, pp. 1087–1089, Oct. 1984.

[20] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis—A unified approach to speech spectral estimation," in *Proc. Int. Conf. Spoken Lang. Process.*, 1994.

[21] R. Sarikaya and J. H. L. Hansen, "Analysis of the root cepstrum for acoustic modeling and fast decoding in speech recognition," *Proc. Eurospeech*, 2001.

[22] C. K. S. Chatterjee and W. B. Kleijn, "Auditory model based optimization of MFCCs inproves automatic speech recognition performance," in *Proc. Interspeech*, Brighton, U.K., 2009.

[23] S. Chatterjee and W. B. Kleijn, "Auditory model based modified MFCC features," in *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, Dallas, TX, 2010, pp. 4590–4593.

[24] Y.-H. Chiu and R. M. Stern, "Analysis of physiologically-motivated signal processing for robust speech recognition," in *Proc. Interspeech*, Brisbane, Australia, Sep. 2008.

[25] M. C. Liberman, "Auditory nerve response from cats raised in a low-noise chamber," *J. Acoust. Soc. Amer.*, vol. 63, pp. 442–455, 1978.

[26] R. F. Lyon, "A computational model of filtering, detection and compression in the cochlea," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Paris, France, May 1982, pp. 1282–1285.

[27] M. G. Heinz, X. Zhang, I. C. Bruce, and L. H. Carney, "Auditory nerve model for predicting performance limits of normal and impaired listeners," *Acoust. Res. Lett. Online*, vol. 2, no. 3, pp. 91–96, 2001.

[28] Y.-H. Chiu, B. Raj, and R. Stern, "Towards fusion of feature extraction and acoustic model training: A top down process for robust speech recognition," in *Proc. Interspeech*, Brighton, U.K., 2009.

[29] Y.-H. Chiu and R. M. Stern, "Learning-based auditory encoding for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, Apr. 2010, pp. 4278–4281.

[30] Y.-H. Chiu and R. M. Stern, "Minimum variance modulation filter for robust speech recognition," in *Proc. IEEE Conf. Acoustics, Speech, Signal Process.*, Taipei, Taiwan, 2009, pp. 3917–3920.

[31] E. Terhardt, "Calculating virtual pitch," *Hear. Res.*, vol. 1, pp. 155–182, 1979.

[32] A. S. Bregman, *Auditory Scene Analysis*. Cambridge, MA: MIT Press, 1990.

[33] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "On a model-robust training method for speech recognition," in *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, Tokyo, Japan, Apr. 1986.

[34] M. A. P. A. Nadas and D. Nahamoo, "On a model-robust training method for speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1432–1436, Sep. 1988.

[35] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, pp. 525–533, 1993.

[36] J. R. Shewchuk, An introduction to the conjugate gradient method without the agonizing pain Comput. Sci. Dept., Carnegie Mellon Univ. , 1994, Tech. Rep. CS-94-125.

[37] A. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones, The NOISEX-92 study on the effect of additive noise on automatic speech recognition DRA Speech Research Unit, Malvern, U.K., 1992, Tech. Rep..

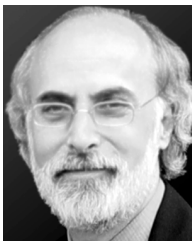
[38] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, pp. 247–251, 1993.

Yu-Hsiang Bosco Chiu (M'10) received the B.S. and M.S. degrees from the Electrical Engineering Department, National Tsing Hua University, Hsinchu, Taiwan, in 2001 and 2003, respectively, and the Ph.D. degree from the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA. in 2010.

His research is in speech recognition and language understanding, where he has focused on the development of automatic learning algorithms for enhancing speech recognition performance under adverse conditions. He is interested in computational perception algorithms that are loosely motivated by physiological principles and that are optimized for best recognition performance.

Bhiksha Raj (M'10) received the Ph.D. degree from Carnegie Mellon University (CMU), Pittsburgh, PA. in 2000.

From 2000 to 2001, he was at Compaq's Cambridge Research Labs, Boston, and from 2001 to 2008 he headed the speech research effort at Mitsubishi Electric Research Labs. Since the fall of 2008, he has been an Associate Professor at the Language Technologies Institute, Carnegie Mellon University, as well as an Associate Professor by Courtesy in CMU's Department of Electrical and Computer Engineering. He has conducted research in a variety of areas including noise robust speech recognition, likelihood-maximizing beamforming, data visualization, and latent-variable spectral decompositions for signal separation. He has also been a major contributor to the Sphinx suite of open-source systems, and he served as the main architect of Sphinx 4. At Mitsubishi, he was primarily responsible for the invention and development of techniques for voice-based search, many of which were highly successful. He holds several patents (and patent applications) in speech recognition, voice search and denoising, and he is the author of over 100 articles in refereed conferences, journals, and books.



Richard M. Stern (M'76) received the B.S. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1970, the M.S. degree from the University of California, Berkeley, in 1972, and the Ph.D. degree from MIT in 1977, all in electrical engineering.

He has been on the faculty of Carnegie Mellon University, Pittsburgh, PA, since 1977, where he is currently a Professor in the Electrical and Computer Engineering, Computer Science, and Biomedical Engineering Departments, and the Language Technologies Institute. Much of his current research is in spoken language systems, where he is particularly concerned with the development of techniques with which automatic speech recognition can be made more robust with respect to changes in environment and acoustical ambience. He has also developed sentence parsing and speaker adaptation algorithms for earlier CMU speech systems. In addition to his work in speech recognition, he also maintains an active research program in psychoacoustics, where he is best known for theoretical work in binaural perception.

Dr. Stern is a Fellow of the Acoustical Society of America and the International Speech Communication Association (ISCA), the 2008–2009 ISCA Distinguished Lecturer, a recipient of the Allen Newell Award for Research Excellence in 1992, and he served as General Chair of Interspeech 2006. He is also a member of the Audio Engineering Society.

Learning-Based Auditory Encoding for Robust Speech Recognition

Yu-Hsiang Bosco Chiu, *Student Member, IEEE*, Bhiksha Raj, *Member, IEEE*, and Richard M. Stern, *Member, IEEE*

***Abstract*—This paper describes an approach to the optimization of the nonlinear component of a physiologically motivated feature extraction system for automatic speech recognition. Most computational models of the peripheral auditory system include a sigmoidal nonlinear function that relates the log of signal intensity to output level, which we represent by a set of frequency dependent logistic functions. The parameters of these rate-level functions are estimated to maximize the *a posteriori* probability of the correct class in training data. The performance of this approach was verified by the results of a series of experiments conducted with the CMU S phinx-III speech recognition system on the DARPA Resource Management, Wall Street Journal databases, and on the AURORA 2 database. In general, it was shown that feature extraction that incorporates the learned rate-nonlinearity, combined with a complementary loudness compensation function, results in better recognition accuracy in the presence of background noise than traditional MFCC feature extraction without the optimized nonlinearity when the system is trained on clean speech and tested in noise. We also describe the use of lattice structure that constraints the training process, enabling training with much more complicated acoustic models.**

***Index Terms*—Auditory model, discriminative training, feature extraction, robust automatic speech recognition.**

I. Introduction

THE human auditory system serves a wide range of functions our daily life, enabling the encoding and recognition of a diversity of environmental sounds such as human speech, animal songs, and background noises. An essential component of this task is the accurate representation of the relative intensity of an incoming sound as a function of frequency. While the method by which the auditory system encodes the intensity of sound is still under debate [1]–[4], one can argue that it is likely to be optimized at some level for the recognition of human speech sounds.

It is often hypothesized that the various aspects of human auditory perception, such as frequency resolution of the cochlea [5], [6], nonlinear compressive effects of the middle ear [7], [8], simultaneous and non-simultaneous masking effects [9]–[12], etc., aid or enhance human ability to recognize speech, particularly in the presence of noise. Researchers have therefore attempted to incorporate many of these attributes into the feature extraction stages of automatic speech recognition systems as well with varying degrees of success (e.g., [13], [14]).

Prior attempts at modeling the human auditory system may broadly be divided into two categories—those that attempt to mimic various aspects of the auditory system, usually through empirically or mathematically derived analytical models of auditory processes (e.g., [15] and [16]), and those that only retain the *framework* of auditory processing, but actually optimize model parameters for automatic speech recognition (e.g., [13], [14]).

The latter approach is particularly attractive for the following reason: it is reasonable to believe that biological auditory processes have been optimized for the manner in which the brain processes and recognizes sounds (subject to other physiological constraints). It is questionable that the detailed structure of human auditory processing is also optimal for automatic speech recognition systems, which are complex statistical machines whose relationship to the actual recognition processes in the brain is unknown. It follows that if we were to optimize the parameters of auditory processing for automatic speech recognition, the resultant feature computation module is likely to result in superior performance compared to features obtained by blind mimicry of auditory models.

Most prior attempts at optimizing the parameters of a physiologically motivated feature computation scheme for automatic recognition have concentrated on the filter bank that is used for frequency analysis. For example, Biem *et al.* propose a discriminative feature extraction procedure which refines the filter bank, by using a smoothed binary loss [17]. Kinnunen used the F-ratio to design a filter bank for improving speaker recognition performance [18]. These methods have primarily addressed data-driven optimization of the frequency analysis of the speech signal. Other authors have attempted to modify the nonlinear compression of feature computation for better speech processing [19] and recognition, e.g., [14], [20]–[23]. Chatterjee *et al.* proposed an augmentation of MFCC features by including higher-order terms of filter bank energy outputs and optimizing them such that the features extracted were similar in terms of the local geometries to the output of auditory model

Manuscript received December 31, 2010; revised April 19, 2011; accepted September 04, 2011. Date of publication September 15, 2011; date of current version nulldate. This work was supported in part by the National Science Foundation under Grants IIS-0420866 and IIS-I0916918, in part by DARPA, and in part by the Charles Stark Draper Laboratory University Research and Development Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark J. F. Gales.

Y.-H. B. Chiu is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15232 USA (e-mail: ychiu@cs.cmu.edu).

B. Raj is with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15232 USA (e-mail: bhiksha@cs.cmu.edu).

R. M. Stern is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15232 USA. He is also with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15232 USA (e-mail: rms@cs.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2011.2168209

[22], [23]. However, their objective is to develop a model that most closely mimics the outputs of actual auditory processing, without particular regard to automatic speech recognition performance.

In this paper, we investigate a technique for the design of yet another physiologically motivated processing stage in feature computation that is optimized for recognition accuracy. In previous work [24], we have determined that the *rate-level nonlinearity* that models the nonlinear relationship between input signal level and the putative rate of firing of fibers of the auditory nerve is a major contributor to robustness in speech recognition. In other physiological studies in cats it has been observed that the distribution of different types of auditory neurons with respect to spontaneous rate of activity depends on the amount of noise in the environment in which the animal was raised [25], indicating that the auditory-nerve response is at least partially a function of the "training" data to which the animal had been exposed. Motivated by these facts, we investigate a technique for automatically learning the parameters of a nonlinear compressive function that mimics the rate-level nonlinearity to optimize recognition accuracy in noise.

We show that we are able to learn a nonlinearity that does indeed improve recognition accuracy significantly in the presence of noise. Additionally, we show that the performance of the learned rate-level nonlinearity has both generalizable and task-specific aspects, validating our hypothesis that the parameters must be *learned* since their optimal values may be different for different tasks.

The rest of this paper is organized as follows. In Section II, we describe the feature computation scheme we will employ, that incorporates a stage modeling the rate-level nonlinearity. In Section III, we describe the learning algorithm to learn the parameters of the nonlinearity. Automatic learning of the parameters can be a computationally expensive process. We also discuss in Section III two ways to reduce the computational complexity associated with the learning process: the use of conjugate gradient descent to reduce the total number of iterations for achieving convergence, and restriction of the gradient search to legal candidate states according to a lattice of allowable word sequences in the training data. In Section IV, we describe experiments conducted on the DARPA Resource Management, Wall Street Journal and AURORA 2 corpora in the presence of several types of noise. Finally, our summary and conclusions are provided in Section V.

II. Feature Computation Using a Rate-Level Nonlinearity and Equal-Loudness Compensation

Most physiologically motivated feature extraction schemes take the form of concatenation of a bank of bandpass filters, a rectifying nonlinearity, and a subsequent additional filtering and other processing components that vary from implementation to implementation (e.g., [15], [16], [26]).

A significant aspect of the human auditory system is a nonlinear relationship between the loudness of perceived sound and neuronal firing rate. Nearly all physiologically motivated feature extraction schemes model this relationship. Typically this is done by a logarithmic or power-law nonlinearity. In the Seneff model in particular [16], this is modeled by a *rate-level* nonlinearity, which operates as a soft clipping mechanism that limits the response to both very small and very large amplitudes of sound.

In a previous study [24], in which we analyzed the contributions of various elements of the Seneff model to speech recognition performance, we determined that the rate-level nonlinearity is the element that provides the greatest robustness with respect to additive noise. The rate-level nonlinearity in auditory models differs from the usual power-law and logarithmic compression used in root-power or mel-frequency cepstra, in that it not only compresses high signal levels, but also low ones. A typical nonlinearity, as abstracted from a model of the peripheral auditory system [27], is shown in the solid curve in the upper left panel of Fig. 1; the dashed curve depicts the traditional logarithmic rate-level nonlinearity used in MFCC and similar processing. Small-amplitude sounds are more easily affected by noise. By nonlinearly compressing small-amplitude signals, the rate-level nonlinearity appears to reduce the effects of noise, resulting in reduced degradation of recognition accuracy.

The lower left panels of Fig. 1 depict separately the amplitude histograms of clean speech in training data, and white noise, with a signal-to-noise ratio (SNR) of 20 dB. Note in these panels that the responses to the speech component are in the graded part of the rate-intensity function while the responses to the less-intense noise fall in the portion of the rate-intensity curve for which the output remains relatively constant independently of the input. In the right panels of the same figure, we show the spectra derived after the traditional log compression (upper right panel) and using the physiologically derived rate-level function (lower left panel). In each case, responses are shown for clean speech and speech degraded by white noise at an SNR of 20 dB, corresponding to the solid and dashed curves, respectively. As can be seen in the figure, the use of the nonlinear rate-intensity function sharply reduces the differences between the shapes of the curves representing clean speech from speech in noise.

As noted above, we argued in [24] that the most important aspect of the auditory model was the nonlinearity associated with the hair cell model. To the extent that this is true, we should be able to obtain a similar benefit by applying such a nonlinearity to conventional MFCC-like feature extraction. Toward this end we modeled the nonlinear curve in the upper left panel of Fig. 1 by a logistic function and interposed it between the log of the triangularly weighted frequency response and the subsequent discrete Fourier transform (DCT) operation in traditional Mel-frequency cepstral coefficient (MFCC) processing, as shown in Fig. 2 ([24], [28]–[30]). Specifically, after windowing the incoming signal into frames of brief duration, a short-time Fourier Transform is applied to obtain the power spectrum of each frame. The power spectrum is then integrated into a Mel-spectrum using traditional triangle-shaped weighting functions to obtain the equivalent of the output of a Mel-frequency filterbank. The filterbank output is then compressed by a logarithmic nonlinearity.

An additional aspect of psychoacoustic models, which we also evaluated as part of the feature computation in [24], is an *equal-loudness weighting* shown in Fig. 3 that is derived

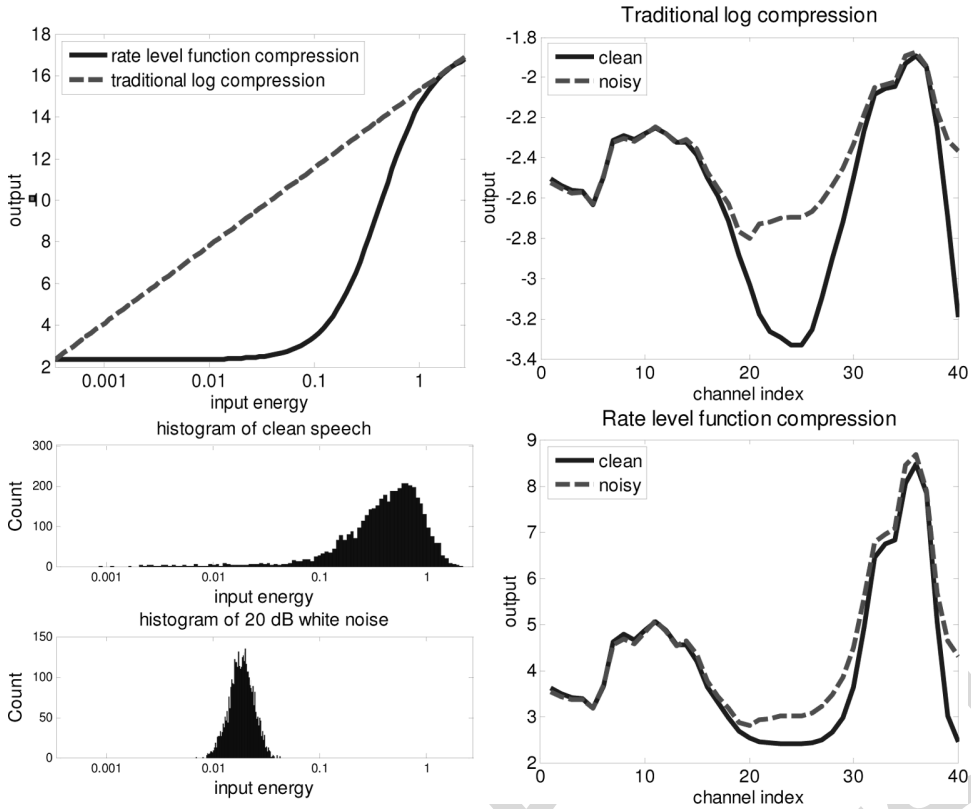


Fig. 1. Upper left panel: physiologically motivated rate-level function in the half wave rectification stage (solid curve) compared with traditional log compression (dashed line). Lower left panels: magnitude (rms) histogram for clean speech and for white noise, with an SNR of 20 dB. Right panels: log Mel spectrum under clean conditions (solid line) and in white noise at an SNR of 20 dB (dashed line). Responses are compared for traditional logarithmic compression and for the rate-level function discussed in this paper (upper and lower right panels, respectively).

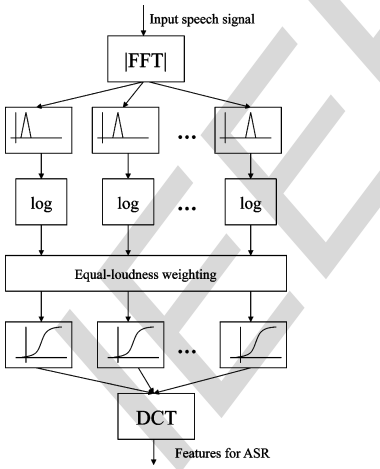


Fig. 2. Block diagram of the feature computation scheme. Note that a frequency weighting and a sigmodial nonlinearity are interposed between the log transformation and the DCT in traditional MFCC processing.

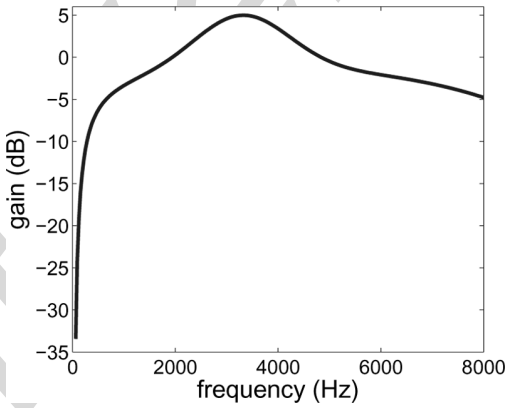


Fig. 3. Function used to approximate equal-loudness weighting based on the results in [31].

from the equal-loudness curve [31] which characterizes psychoacoustical results relating signal intensity to perceived loudness. While in reality perceived loudness depends on both the frequency and intensity of the incoming signal, we only normalize the mean response and assume that it is dependent only on frequency.

In computational models, equal-loudness weighting is implemented as a constant, frequency-dependent multiplicative weighting of the filter-bank output. In our implementation we apply it instead as an additive correction to the logarithm-compressed mel-frequency filterbank output, which is why the equal-loudness weighting appears after the log operation in Fig. 2.

The equal-loudness weighted log-mel-spectrum is then passed through a logistic function that is introduced to model the nonlinear average auditory-nerve response as a function of the input level in decibels

$$x_{u,t}[n] = \frac{\alpha[n]}{1 + exp(w_1[n] \cdot y_{u,t}[n] + w_0[n])} \qquad (1)$$

where $y_{u,t}[n]$ is the nth log Mel-spectral value and $x_{u,t}[n]$ is the corresponding sigmoid-compressed value of frame t in utterance u. The parameters of the nonlinearity, $\alpha[n] = 0.05$; $w_0[n] = 0.613$; $w_1[n] = -0.521$, $\forall\, n$, were determined empirically by evaluation on the Resource Management development with white noise added at an SNR of 10 dB. These parameter values are used in all our experiments. Note that these values are the same for all Mel-frequency components, i.e., they are frequency independent. Finally, cepstral-like coefficients were obtained by applying the DCT transform to the output of the rate-level nonlinearity.

A final note on equal-loudness weighting: in conventional feature computation that employs a logarithmic nonlinearity, the equal-loudness weighting is canceled out by the cepstral mean subtraction (CMS) that is routinely used in speech recognition. This is one reason why it is generally not used in Mel-frequency cepstral computation, but remains a part of some other feature computation schemes such as PLP, which use other forms of compression. In our model too, logistic compression following the logarithmic compression ensures that the equal-loudness weighting is not canceled out by CMS.

III. Learning the Nonlinearity

The premise of our paper is that the nonlinearities in the human auditory tract are a function of more than mere recognition performance (a view endorsed by Bregman [32] among others), but nevertheless serves a demonstrably useful purpose in recognition. In a computational model that need not consider other factors not related to recognition, the principle behind the nonlinearity could be retained, while the actual form with which it is implemented could be explicitly optimized for recognition performance.

Rather than hypothesize an entirely new form for the nonlinearity however, we retain the sigmoidal form described in (1), but attempt to determine the *parameters* of the nonlinearity to optimize recognition accuracy obtained with an automatic speech recognition system.

Unfortunately, the hidden Markov models for the various phonemes and the language model used for automatic speech recognition are quite complex, and it is difficult to obtain a simple update mechanism that can relate recognition accuracy to the parameters of the sigmoidal nonlinearity. Because of this, we use a simple Bayesian classifier for sound classes in the language as a substitute for the recognizer itself. Each sound class is modeled by a Gaussian distribution, computed from training data for that sound class. We use a maximum-mutual information (MMI) criterion to estimate the parameters of the nonlinearity such that the posterior probabilities of the phonemes based on their own training data are maximized.

The basic formulation for MMI training [33], [34] is well known. Given a parametric model $P(\mathbf{s}, C; \theta)$ expressing the joint probability distribution of data $\mathbf{s}$ and a class label C with parameters θ, MMI training learns θ such that the mutual information $I_\theta(\mathbf{s}, C)$ is maximized. The mutual information between $\mathbf{s}$ and C is given by

$$I_\theta(\mathbf{s}, C) = \log(\frac{P(\mathbf{s}, C)}{P(C)P(\mathbf{s})}) = \log P(C|\mathbf{s}) - \log P(C). \quad (2)$$

In the above equation, we have not explicitly represented θ, with the understanding that it represents the set of parameters of the model. Thus, for a given $P(C)$, maximizing the mutual information is equivalent to maximizing $\log P(C|\mathbf{s})$, where *a posteriori* probability is given by the usual Bayesian decomposition

$$P(C|\mathbf{s}) = \frac{P(\mathbf{s}|C)P(C)}{\sum_{C'} P(\mathbf{s}|C')P(C')} = \frac{P(\mathbf{s}|C)}{\sum_{C'} P(\mathbf{s}|C')} \quad (3)$$

with equal prior probabilities assigned to all sound classes. In our problem, C represents the sound classes, $\mathbf{s}$ represents the set of sequences of feature vectors for the recordings in our training set, i.e., $\mathbf{s} = \{\mathbf{s}_{u_1}, \mathbf{s}_{u_2}, \ldots\}$, where $\mathbf{s}_u$ is the sequence of feature vectors for any utterance u in our training set, and $\mathbf{s}_u = [\mathbf{s}_{u,1}, \mathbf{s}_{u,2}, \ldots]$ where $\mathbf{s}_{u,t}$ is the tth feature vector in $\mathbf{s}_u$.

We will also make use of the following approximations. We assume that individual utterances u are mutually statistically independent. We also assume that the *a posteriori* probability of the *true* label for an utterance, $C_u = \{C_{u,t},\ t = 1 \cdots T\}$ (individual vectors in the utterance may have different labels) is the product of the *a posteriori* probability of the individual vectors

$$\begin{aligned} P(C_u|\mathbf{s}_u) &= P(C_u)\frac{P(\mathbf{s}_u|C_u)}{\sum_{C'_u} P(\mathbf{s}_u|C'_u)P(C'_u)} \\ &\approx P(C_u)\prod_{t=1}^{T}\frac{P(\mathbf{s}_{u,t}|C_{u,t})}{\sum_{C'_{u,t}} P(\mathbf{s}_{u,t}|C'_{u,t})P(C'_{u,t})}. \end{aligned} \quad (4)$$

In other words, we assume that $\log P(C_u|\mathbf{s}_u) = \sum_{t=1}^{T} \log P(C_{u,t}|\mathbf{s}_{u,t})) + \log P(C_u)$. This approximation, which actually occurs in the denominator of the last term in the equation (which must ideally be summed over all class and HMM-state sequences), ignores the dependencies between class labels of adjacent vectors. It also ignores the contributions of the transition probabilities of the HMMs. These approximations greatly enhance the tractability of the problem, as explicitly incorporating these dependencies would greatly complicate the optimization. We also observed in pilot studies that including the transition probabilities into the estimation did not enhance the performance of the algorithm.

The actual optimization is performed using gradient descent. This is illustrated by Fig. 4.

The procedure for optimizing the nonlinearity is as follows. We assume we have a collection of training recordings: $\mathbf{s} = \{\mathbf{s}_{u_1}, \mathbf{s}_{u_2}, \ldots\}$ and their true labels $C = \{C_{u_1}, C_{u_2}, \ldots\}$.

Let $\boldsymbol{\mu}_C$ be the mean vector and $\boldsymbol{\sigma}_C$ be the covariance of the feature vectors for any sound class C. The likelihood of any vector $\mathbf{s}_{u,t}$, as computed by the distribution for that sound class is assumed to be given by a Gaussian density $N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})$. Further, we also assume that the individual classes $C_{u,t}$ are equally likely. This assumption not only simplifies our computation; in practice it was not observed to affect our results. The posterior probability of any sound class $C_{u,t}$, given a specific observation $\mathbf{s}_{u,t}$ is given by

$$P(C_{u,t}|\mathbf{s}_{u,t}) = \frac{N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})}{\sum_{C'} N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})} \quad (5)$$

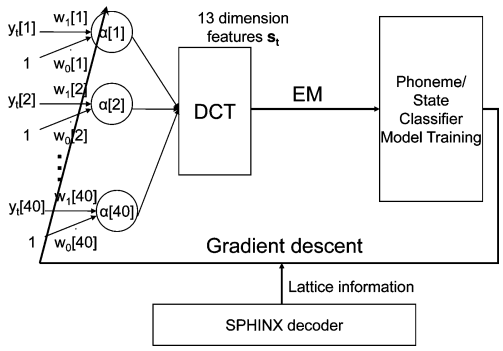


Fig. 4. Integrated system that refines the parameters characterizing the rate-level nonlinearity using the accuracy of a simple phonetic classifier as the objective function.

under the assumption that the prior probabilities of each class are equal.

The total overall log *a posteriori* probability of the true labels C of $\mathbf{s}$ is given by

$$\log P(C|\mathbf{s}) = \log P(C) + \sum_u \sum_t \log \frac{N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})}{\sum_{C'} N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})} \tag{6}$$

where u sums over all utterances and t sums over all features vectors for each utterance. Thus, optimizing $\log P(C|\mathbf{s})$ is equivalent to optimizing $\sum_u \sum_t \log(N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})/\sum_{C'} N(\mathbf{s}_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'}))$.

Our objective is to estimate the parameters of the sigmoidal nonlinearity of (1) to optimize $\log P(C|\mathbf{s})$ (for brevity, we will simply refer to $\log P(C|\mathbf{s})$ as $\log(P)$ henceforth). In doing so, however, it must also consider other aspects of the computation. Cepstral mean subtraction is a common component of speech recognition systems and is employed by us. The optimization algorithm must take this into consideration. In other words, we will actually optimize

$$\log(P) = \sum_u \sum_t \log \frac{N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_{C_{u,t}}, \boldsymbol{\sigma}_{C_{u,t}})}{\sum_{C'} N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})} \tag{7}$$

where $\mathbf{s}^c_{u,t}$ is the *mean normalized* feature vector: $\mathbf{s}^c_{u,t} = \mathbf{s}_{u,t} - T_u^{-1} \sum_t \mathbf{s}_{u,t}$, where T_u is the number of features vectors in the utterance u. Here we have also ignored $\log P(C)$ as being irrelevant to our algorithm.

Also, modifying the manner in which features are computed will also modify the Gaussian distributions of the classes. Hence, the parameters of the Gaussian distributions of each sound class, and those of the sigmoidal nonlinearity in the feature computation, are jointly estimated to maximize $\log(P)$.

A. Estimating the Sound-Class Distribution Parameters

The model parameters $\boldsymbol{\mu}_C$ and $\boldsymbol{\sigma}_C$ for each sound class are initialized by training HMMs for all sound units using the conventional Baum–Welch algorithm, and conventional MFCC features. Thereafter, they are updated using the same objective criterion employed by the speech recognizer. For maximum-likelihood training, this is given by

$$\begin{aligned}
\boldsymbol{\mu}_C &= \frac{1}{\sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C)} \sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C) \mathbf{s}^c_{u,t}, \\
\boldsymbol{\sigma}_C &= \frac{1}{\sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C)} \\
&\quad \cdot \sum_u \sum_t I(\mathbf{s}^c_{u,t} \in C) \left(\mathbf{s}^c_{u,t} - \boldsymbol{\mu}_C\right) \left(\mathbf{s}^c_{u,t} - \boldsymbol{\mu}_C\right)^T
\end{aligned} \tag{8}$$

where $I(\mathbf{s}^c_{u,t} \in C)$ is an indicator function that takes a value of 1 if $\mathbf{s}^c_{u,t}$ belongs to sound class C and 0 otherwise.

B. Estimating the Parameters of the Sigmoidal Nonlinearity

The parameters for the logistic function $\mathbf{F} = \{\boldsymbol{\alpha}, w_0, w_1\}$ are estimated by maximizing $\log(P)$ using a gradient descent approach. Taking the derivative of the objective function with respect to $\mathbf{F}$, the nonlinear parameters are updated as

$$\begin{aligned}
\boldsymbol{\alpha}^{\text{new}} &= \boldsymbol{\alpha}^{\text{old}} + 0.001 \cdot \gamma \cdot \frac{\partial \log P}{\partial \boldsymbol{\alpha}} \\
\boldsymbol{w}_0^{\text{new}} &= \boldsymbol{w}_0^{\text{old}} + \gamma \cdot \frac{\partial \log P}{\partial \boldsymbol{w}_0} \\
\boldsymbol{w}_1^{\text{new}} &= \boldsymbol{w}_1^{\text{old}} + 0.2 \cdot \gamma \cdot \frac{\partial \log P}{\partial \boldsymbol{w}_1}.
\end{aligned} \tag{9}$$

The forms of the partial derivatives are provided in Appendix. The weighting terms 0.001 and 0.2 were empirically obtained factors intended to result in roughly equal convergence rates for all three parameters, and the step size γ is equal to 0.05 in our experiments.

Our objective is to derive sigmoidal parameters minimizing the distortion in the features that results from corruption by noise. Thus, while class distribution parameters are learned from clean data, the sigmoidal parameters are learned to optimize classification on both clean and noisy data.

Thus, the updates of (9) are performed on both clean and noisy data, whereas the model updates of (8) are performed on clean data. After each step of gradient descent according to (9), the model parameters are updated using (8) on the clean training set only.

The procedure is iterative. Finally, once the objective function $\log(P)$ has converged, the nonlinearity parameters $\mathbf{F} = \{\boldsymbol{\alpha}, \boldsymbol{w_0}, \boldsymbol{w_1}\}$ are retained for the feature extraction process.

The model parameters of the entire speech recognition system are then retrained using features derived using the learned nonlinearity from the clean training set.

The entire learning algorithm is described in Algorithm 1. Here $\mathbf{y}_{u,t}$ represents the set of log mel-spectra that are input to the sigmoidal nonlinearity in (1) and $\mathbf{F}$ represents the set of parameters for the sigmoidal nonlinearity, as mentioned earlier. The feature vector $\mathbf{s}_{u,t}$ is derived from $\mathbf{y}_{u,t}$ as

$$\mathbf{s}_{u,t} = DCT(sigmoid(\mathbf{y}_{u,t}, \mathbf{F})) \tag{10}$$

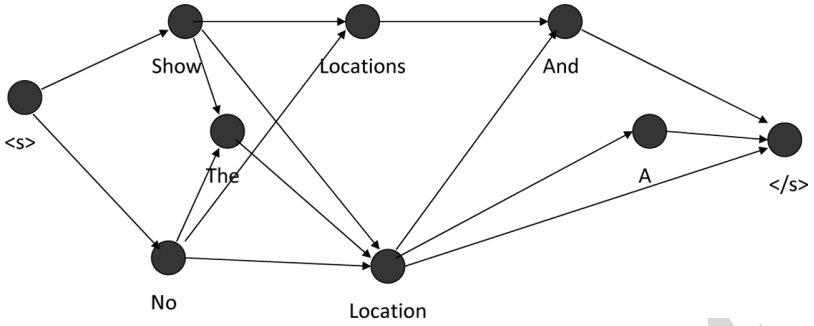


Fig. 5. Example of a word lattice to reduce the computational complexity by including only decoder-identified candidates as the competing classes.

as illustrated in Fig. 2, where $sigmoid(\mathbf{y}_{u,t}, \mathbf{F})$ represents the sigmoidal function of (1). Note that the sigmoid is applied individually to every spectral component in each log-mel-spectral vector y. Also, the sigmoidal parameters $\mathbf{F}$ are different for individual Mel-spectral channels. In addition, although Algorithm 1 only explicitly requires the features and does not require them to be mean normalized, the derivatives used in the update (given in the Appendix) are actually computed from mean-normalized features, thus explicitly accounting for CMS.

Algorithm 1: Algorithm for learning the parameters of the sigmoidal nonlinearity.

Input: $\mathbf{F}$, $\{\mathbf{y}_{u,t}, C_{u,t},\ u = 1 \ldots U, t = 1 \ldots T_u\}$

Output: $\mathbf{F}$

while *not converged* **do**

1. Compute the feature vector $\{\mathbf{s}_{u,t} \ \forall \ u,t\}$ from $\{\mathbf{y}_{u,t} \ \forall \ u,t\}$ using $\mathbf{F}$ in **(10)**.

2. Estimate $\{\boldsymbol{\mu}_C, \boldsymbol{\sigma}_C\} \ \forall \ C$ using **(8)** on the clean training set

3. Compute $\log(P)$ using **(7)** on both the clean and noisy training sets

4. $\mathbf{F}_{\text{new}} \leftarrow \mathbf{F}_{\text{old}} + \partial \log P / \partial \mathbf{F}$ using **(9)** on both clean and noisy training set

end

At convergence, the algorithm learns the optimal sigmoidal parameters $\mathbf{F}$ for each Mel-spectral channel.

C. Reducing Computational Complexity by Using a Word Lattice

A complete MMI solution that computes the ratio of the probability of the "true" class label to the sum of the probabilities of *all* classes can become prohibitively computationally expensive. The solution in the previous section assumes that each class is modeled by a single Gaussian. It is straightforward to extend it to include a mixture of M Gaussians (although, for pragmatic reasons we have not done so as we will explain in the concluding section of the paper). The amount of computation for calculating derivatives for each set of parameters at each iteration will be on the order of $O(KLMN)$, where K is the number of cepstral dimensions, L is the number of channels, and N is the number of sound classes. As the number of Gaussians increases with the complexity of the speech recognizer, the amount of computation becomes too large for feasible implementation.

The key reason for this computational explosion is the denominator in (7). For every observation, we must sum over all classes. To overcome this problem, we restrict the set of "competing" classes for each vector using a word lattice as shown in Fig. 5. Only the classes present in the word lattice are included in the MMI updates for any class: for each feature vector the set of competing classes that are considered when computing the *a posteriori* probability of the true class are only those classes that are present in the lattice at the same time instant as the vector. This affects the computation of both $\log P$ in (13) and the derivative of $logP$ in (14), in that the $\sum_{C'}$ becomes $\sum_{C'_{u,t} \in \text{activeclasses of } \mathbf{s}^c_{u,t} \text{ in the lattice}}$ in both equations. This results in a significant reduction of the number of competitors to be considered, and thereby the overall computation.

Using the word lattice in this manner also has a second effect. The lattice is obtained by recognizing the utterance using our initial acoustic models, along with a language model. The lattice hence represents the *a posteriori* most likely sequences of class labels, and thus implicitly factors in the distribution over class sequences into the objective function of (7), instead of simply marginalizing it out as (7) does.

In our experiments the word lattices were generated using the CMU Sphinx decoder on the training data using the initial acoustic model parameters. The lattices were saved and subsequently remained fixed throughout the optimization process.

D. Optimizing the Speed of Convergence Using Conjugate Gradient Descent

It is well known that the simple gradient-optimization approach, such as that followed in (9) tends to be slow: the gradients at consecutive iterations tend to have high correlation, as a result of which the steps taken at consecutive iterations are very similar and are somewhat redundant. The method of conjugate gradient descent [35], [36] avoids this problem by ensuring that the steps taken at consecutive iterations are orthogonal to one another in the parameter space. This dramatically increases the speed with which the solution is obtained.

We therefore modified our basic algorithm to implement the method of conjugate gradient descent. The modified algorithm is summarized in Algorithm 2. In the algorithm M is a $3L \times 3L$ diagonal weighting matrix, where L is the number of mel-frequency components in each mel-log-spectral vector. The diagonal entries of M are the weights used in (9)—the first L entries are 0.001, the next L are 1, and the final L diagonal entries are 0.2.

Algorithm 2: Algorithm for learning the parameters of the sigmoidal nonlinearity where $\sigma_0 = 0.05$ and $j_{max} = 5$. The matrix M is diagonal with entries equal to $[0.001, \ldots, 1, \ldots, 0.2, \ldots,]$, which are the weights used in **(9)**. The variable r represents the raw gradient, s is the scaled gradient, d is the conjugate gradient search direction, and β measures the projection of the previous search direction onto the current search direction. The inner loop performs j_{max} iterations of a line search in the search direction. The outer loop updates the search direction to a new orthogonalized gradient, or, if the projection is negative, to a new scaled gradient.

Input: $\mathbf{F}$, $\{\{\boldsymbol{y}_{u,t}, C_{u,t}\},\ u = 1 \ldots U, t = 1 \ldots T_u\}$

Output: $\mathbf{F}$

1. $r \leftarrow \partial \log P / \partial \mathbf{F}$ as developed in **Appendix**

2. $s \leftarrow Mr$ where M is a weighting matrix representing the weighting shown in **(9)**

3. $d \leftarrow s$

4. $\delta_{\text{new}} \leftarrow r^T d$

5. **while** *not converged* **do**

6. $j \leftarrow 0$

7. $\gamma \leftarrow \sigma_0$

8. **while** $j < j_{max}$ **do**

9. Compute feature vector $\{\mathbf{s}_{1,1}, \ldots, \mathbf{s}_{u,T_u}\}$ using **(10)**

10. Estimate $\{\boldsymbol{\mu}_C, \boldsymbol{\sigma}_C\}\ \forall\ C$ on the clean training set

11. Compute $\log(P)$ using **(7)** on both clean and noisy training set

12. $\eta \leftarrow [\partial \log P / \partial \mathbf{F}]^T d$

13. **if** $j \neq 0$ then

14. $\gamma \leftarrow \gamma(0.5\eta/\eta' - \eta)$

end

15. $\mathbf{F}_{\text{new}} \leftarrow \mathbf{F}_{\text{old}} + \gamma d$

16. $\eta' \leftarrow \eta$

17. $j \leftarrow j + 1$

end

18. $r \leftarrow \partial \log P / \partial \mathbf{F}$

19. $\delta_{\text{old}} \leftarrow \delta_{\text{new}}$

20. $\delta_{\text{mid}} \leftarrow r^T s$

21. $s \leftarrow Mr$

22. $\delta_{\text{new}} \leftarrow r^T s$

23. $\beta \leftarrow \delta_{\text{new}} - \delta_{\text{mid}} / \delta_{\text{old}}$

24. **if** $\beta \leq 0$ **then**

25. $d \leftarrow s$

else

26. $d \leftarrow s + \beta d$

end

end

IV. Experimental Results

Experiments were run on the DARPA Resource Management RM1 and the DARPA Wall Street Journal WSJ0 corpora to evaluate the methods that are proposed above. The Sphinx-III continuous-density HMM-based speech recognition system was used in all experiments. The feature extraction employed a 40-filter Mel filter bank covering the frequency range of 130 to 6800 Hz. Each utterance is normalized to have zero mean and unit variance before multiplication by a 25.6-ms Hamming window with frames updated every 10 ms.

A. Effect of Frequency Equalization

In our system implementation, each log spectral component is shifted by the equal loudness function shown in Fig. 3. As we mentioned before, this linear filtering does not affect the performance of traditional MFCC processing as it introduces an additive constant to the cepstral coefficients that is removed by cepstral mean subtraction (CMS). In contrast, the sigmoidal nonlinearity affects the frequency normalization in a nonlinear fashion and therefore it is not eliminated by CMS. To better understand the effect of gain in our feature extraction system, we compare system performance of our system with and without the frequency-normalization component.

The feature extraction scheme described in Fig. 2 was applied to utterances from the DARPA Resource Management RM1 database which consists of Naval queries. 1600 utterances from the database were used as our training set and 600 randomly selected utterances from the original 1600 testing utterances were used as our testing set. 72 speakers were used in the training set and another 40 speakers in the testing set, representing a variety of American dialects. We used CMU's SPHINX-III speech recognition system with 1000 tied states, a language model weight of 9.5 and phonetic models with eight Gaussian mixtures. Cepstral-like coefficients were obtained for the proposed system by computing the DCT of the outputs of the nonlinearity. The major difference between traditional MFCC processing and our present approach (both with and without the frequency weighting) is in the use of the rate-level nonlinearity described above. Cepstral mean subtraction (CMS) was applied, and delta and delta-delta cepstral coefficients were developed in all cases in the usual fashion. The parameters of the nonlinearity are $\alpha[n] = 0.05$; $w_0[n] = 0.613$; $w_1[n] = -0.521$, $\forall\ n$ as was mentioned in Section II.

Recognition experiments were run on speech corrupted by a variety of noises. The noises were obtained from the NOISEX-92 database (including a later release of NOISEX) [37], [38], and included recordings of speech babble, and real noise samples in a market, restaurant and theater. All of these noises are digitally added to the original clean test set at SNRs of 0, 5, 10, 15, and 20 dB. We plot recognition accuracy, which is computed as 100% minus the word error rate, where the latter is defined to be the ratio of the total number of insertion, deletion, and substitution errors divided by the number of incoming words.

Fig. 6 compares speech recognition accuracy of the proposed system with and without the equal loudness curve in the presence of four different types of background noise. The horizontal

axis represent the SNR of the test set and vertical axis represents recognition accuracy (calculated as 100%—WER). The filled squares and diamonds represent the recognition accuracy obtained using the rate-level nonlinearity with and without out equal loudness weighting, respectively, and the triangles and open triangles represent the same index using traditional MFCC processing. As can be seen from the figure, the equal loudness curve (which can be thought of as a manipulation of the parameter $w_0[n]$ in different frequency channels) substantially improves speech recognition accuracy, especially in natural environments such as the market or a theater when the rate-level nonlinearity is used. Frequency weighting has almost no impact on the performance of traditional MFCC processing, as expected. We will discuss the optimal parameter values in greater depth below.

B. Recognition Accuracy Using Optimized Nonlinear Parameters

Our sigmoidal rate-level nonlinearity is trained on clean speech from the RM1 database to which pink noise from the NOISEX-92 corpus was digitally added at an SNR of 10 dB. Class labels for training were based on an HMM with 1000 tied states that was generated by forced alignment of the clean training data using previously trained models. The noisy testing sets were created by artificially adding babble noise from the NOISEX-92 corpus, the recordings of market, theater and restaurant noises obtained in real environments to the original clean testing set. We note that the noises used in these training and testing environments were different. The step size was set to 0.05 to achieve stable but reasonably fast convergence.

The choice of 10-dB pink noise was based on preliminary experiments performed on a held-out data set [30]. In general, we found that as long as the energy distribution of the spectrum of the noise used for training is similar to that of the noise in the test data (e.g., the power spectrum decreases from low frequencies to high frequencies), the actual type of noise used for training does not matter. The actual SNR chosen is also supported by past experience: 10 dB tends to be close to the "knee" in plots of recognition error as a function of SNR. If the recognition performance at this noise level is improved, overall performance in the presence of noise tends to improve as well.

Fig. 7 shows the rate-level nonlinearities that were actually learned. Fig. 7(a) is a 3-D plot showing the nonlinearities for all 40 Mel-frequency channels. Fig. 7(b) depicts a few cross-sections of this plot. Fig. 7(c)–(e) show how the individual parameters of the rate-level nonlinearities vary as a function of frequency. We note that the estimated optimal rate-level functions vary greatly across frequencies in all aspects, including gain, slope, and attack.

In comparing the rate-level functions that are learned for different types of background noise, we have found that while the details of the resulting functions differ slightly, the general trends are similar, with a shallow slope in the middle to capture the large dynamic range of speech frequency components in the mid frequencies and a steeper slope in both the low- and high-frequency regions.

Once the parameters of the feature computation module were learned, the feature computation module was employed to derive features from a clean version of the RM training set, from which the HMM model parameters were retrained.

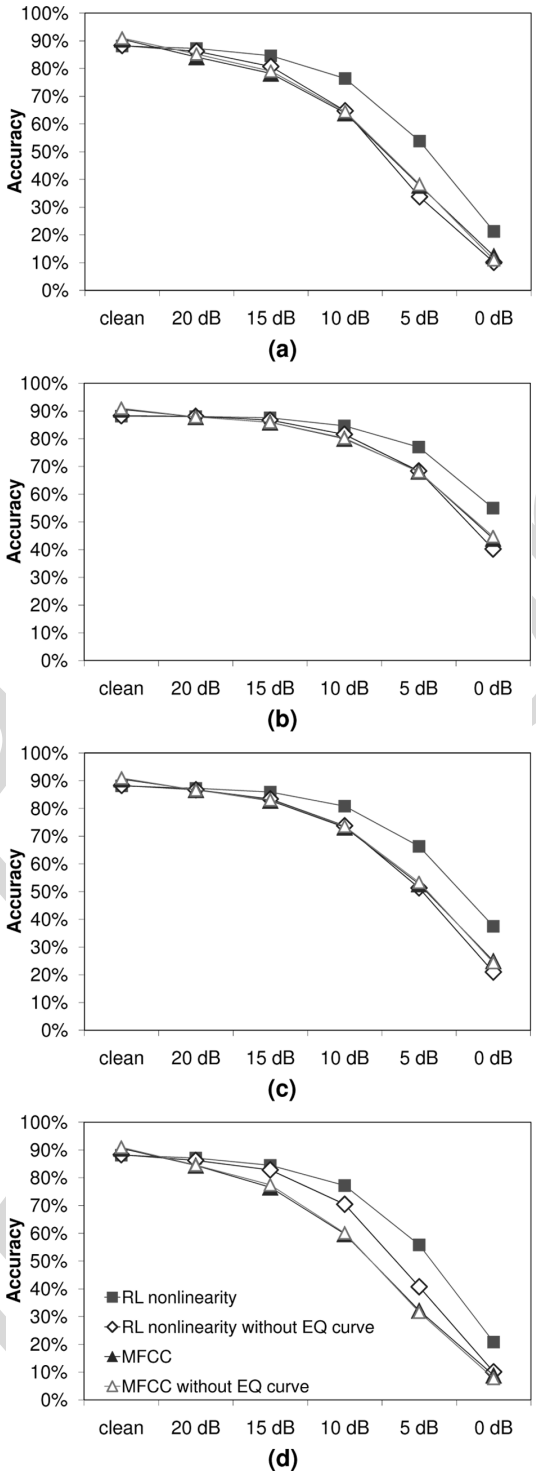


Fig. 6. Demonstration of the impact of the rate-level (RL) nonlinearity and the equal loudness curve. System recognition accuracy (100%—WER) is compared using the RL nonlinearity with equal loudness weighting (squares), the same system without equal loudness weighting (diamonds), baseline MFCC processing (triangles), and baseline MFCC processing without equal loudness weighting (empty triangles) for the RM database in the presence of four different types of background noise. The WERs obtained training and testing with clean speech are—RL nonlinearity: 11.88%, RL nonlinearity without weighting: 11.72% MFCC: 9.45%, MFCC without weighting: 9.07%. (a) Babble noise. (b) Market noise. (c) Restaurant noise. (d) Theater noise.

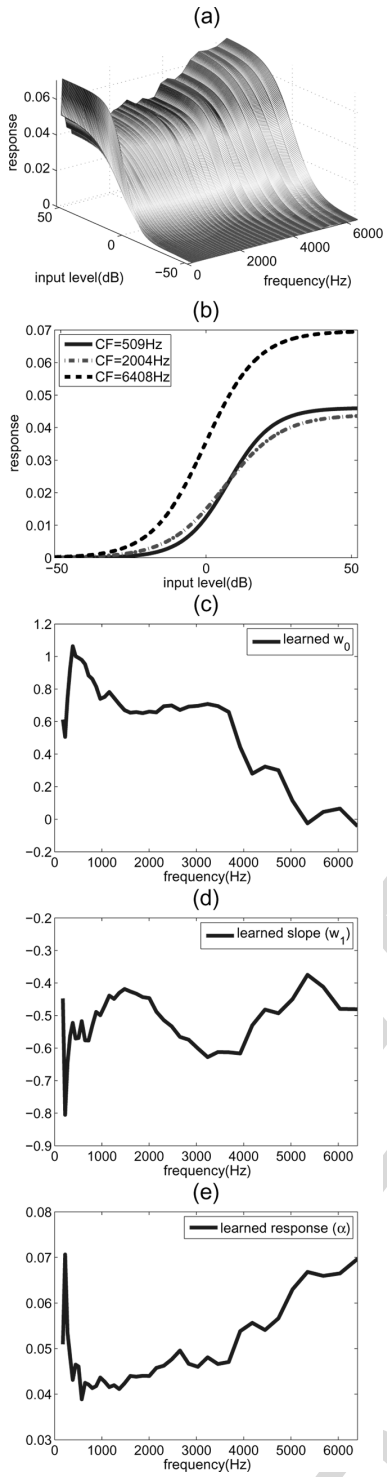


Fig. 7. (a) Trained RL nonlinearity as a function of input intensity and frequency. (b) Examples of the trained RL nonlinearity at selected low, mid and high frequencies. (c)–(e) The trained values of the logistic function parameters w_0, w_1, and α as a function of frequency.

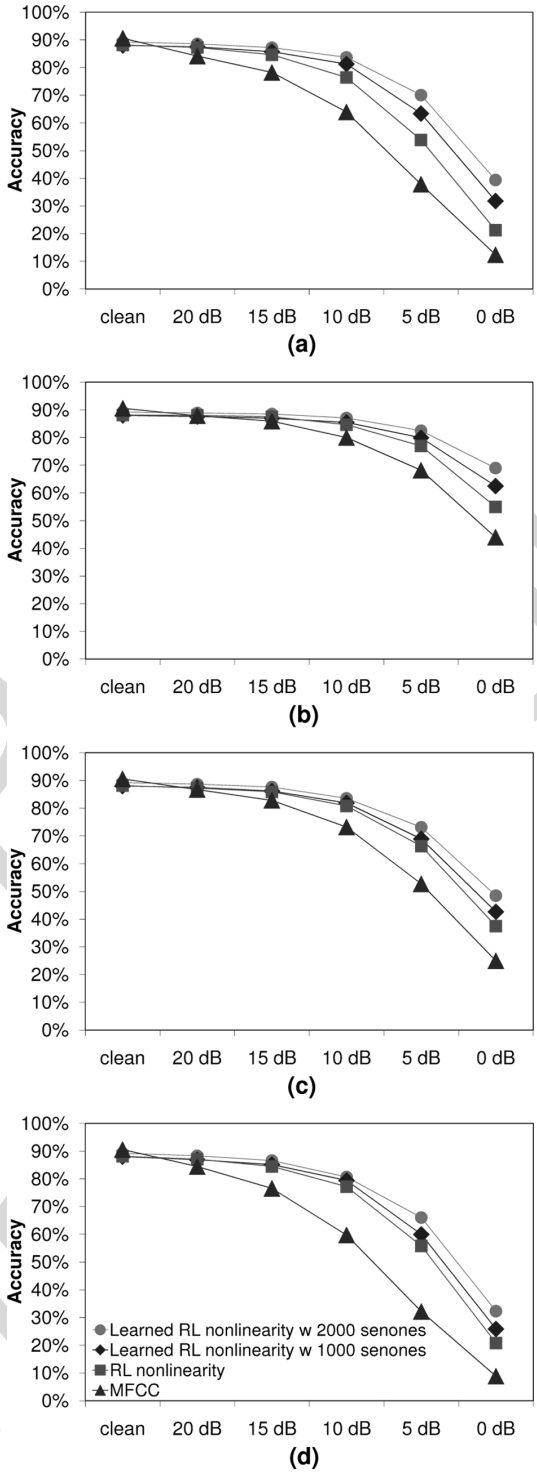


Fig. 8. Comparison of the recognition accuracy obtained using the optimized RL nonlinearity to that obtained with the baseline RL nonlinearity (without optimization) and with MFCC coefficients, all in the presence of four types of background noise using the RM corpus. The WERs obtained training and testing with clean speech are—MFCC: 9.45%, RL nonlinearity: 11.88% with p-value compared to MFCC: 0.00003, RL nonlinearity learned from 1000 tied states: 11.97%, p-value: 0.000013, RL nonlinearity learned from 2000 tied states: 10.53%, p-value: 0.055. (a) Babble noise. (b) Market noise. (c) Restaurant noise. (d) Theater noise.

Fig. 8 compares the recognition accuracy that was obtained using the optimized rate-level nonlinearity (training with 1000 and 2000 senones) to the corresponding results obtained using a similar nonlinearity derived from a model of physiological data [27] with some empirical tuning but no systematic optimization [30], and conventional MFCC coefficients. We note that

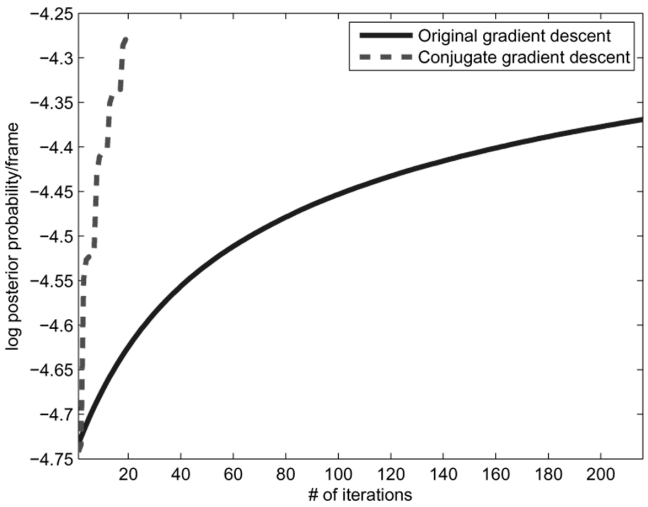


Fig. 9. Comparison of the log posterior probability as a function of the number of iterations using conjugate gradient descent (dashed curve) and traditional gradient descent (solid curve).

while the use of the frequency weighting and rate-level nonlinearity from physiological measurements greatly improves noise robustness compared to the MFCC baseline *(with accuracy loss under clean condition)*, the best results are obtained with the automatically learned parameters. As before, the noise types used for training and testing are different in these experiments.

C. Improvements to Training Speed

Improvement in training speed is provided by two major factors: a reduction in the time required for each iteration of the training procedure (through the use of the lattice representation described in Section III-C), and a reduction in the total number of iterations over the entire training process (which is provided by the use of conjugate gradient descent, as described in Section III-D).

The use of the lattice structure as described in Section III-C reduces the processing time per iteration by reducing the number of competing candidate hypotheses that need to be considered. In empirical comparisons of the processing time with and without the lattice representation we observed that the use of the word lattice reduces the processing time for each iteration of the gradient descent from an average of 727 to 291 s, a reduction factor of approximately 2.5.

Fig. 9 compares the *a posteriori* probability as a function of the number of iterations needed to achieve convergence. The dashed line describes convergence using the conjugate gradient descent method while the solid line describes convergence using the traditional gradient descent method, using the Resource Management database. As can be seen from the figure, the use of conjugate gradient descent reduces the number of iterations required to achieve convergence by a factor of at least 10 compared to traditional gradient descent. We have observed empirically that the performance of the recognition system approximates its optimal value after about 20 iterations of training.

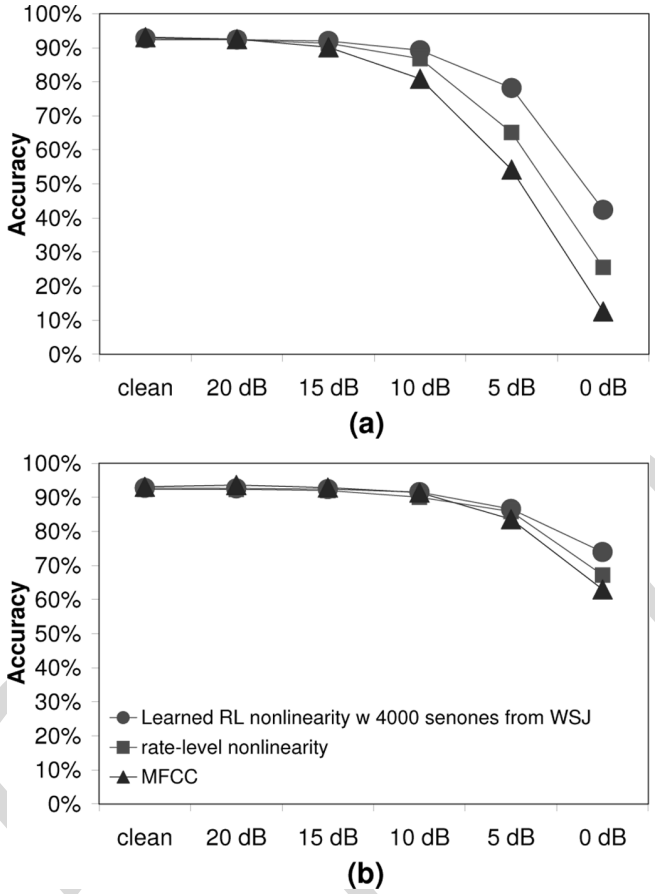


Fig. 10. Comparison of recognition accuracy in the presence of two types of background noise on the WSJ corpus, using procedures similar to those in Fig. 8. The WERs obtained training and testing with clean speech are—MFCC: 6.91%, RL nonlinearity: 7.66%, p-value compared to MFCC: 0.135, RL nonlinearity using 4000 tied states: 7.25%, p-value: 0.493. (a) Babble noise. (b) Market noise.

D. Recognition Accuracy Using the DARPA Wall Street Journal Database

We also evaluated our recognition system using the sigmoidal rate-level nonlinearity with optimized parameters on the standard DARPA Wall Street Journal (WSJ) database. The training set we used consisted of 7024 speaker-independent utterances from 84 speakers. The test set consisted of 330 speaker-independent utterances from the evaluation set of the 5000-word WSJ0 database, using non-verbalized punctuation. As with the Resource Management database, a noisy test set was created by artificially adding babble noise from the NOISEX-92 database and market noise from recordings in real environments at pre-specified SNRs of 0, 5, 10, 15, and 20 dB. The noisy training set was created by adding 10-dB pink noise from the NOISEX-92 database to the original clean training set. The SPHINX-III trainer and decoder were implemented using 4000 tied states, a language model weight of 11.5, and 16 components in all GMMs, with no further attempt made to tune system parameters. Other conditions are the same as in the RM case.

Fig. 10 shows results using the WSJ database for conditions that are similar to those depicted in Fig. 8 except that only a subset of testing noises are examined and a greater number of

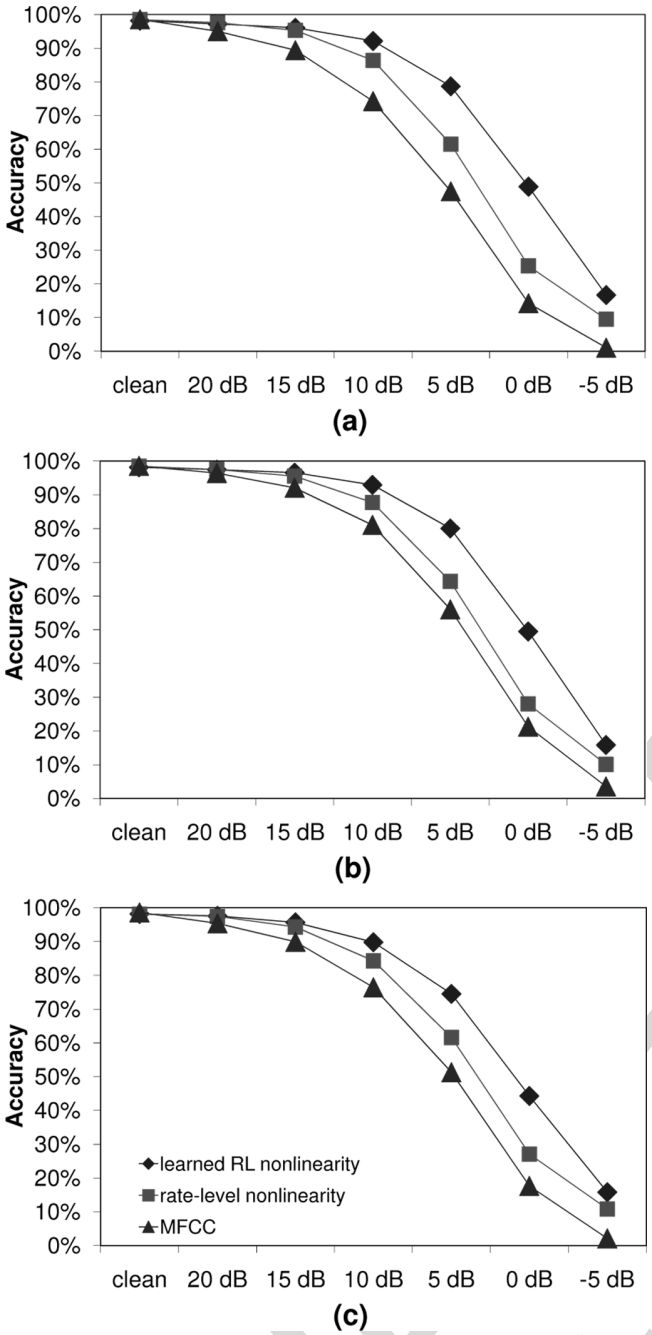


Fig. 11. Comparison of recognition accuracy in the presence of three sets of background noise from the AURORA 2 corpus. The WERs obtained training and testing with clean speech are MFCC: Test A 1.43%, Test B 1.43%, Test C 1.42%,RL nonlinearity: Test A 1.54%, p-value compared to MFCC: 0.461, Test B 1.54%, p-value: 0.461, Test C 1.93%, p-value: 0.023, learned RL nonlinearity: Test A 1.86%, p-value: 0.006, Test B 1.86%, p-value: 0.006, Test C 1.86%, p-value: 0.047. (a) Test set A. (b) Test set B. (c) Test set C.

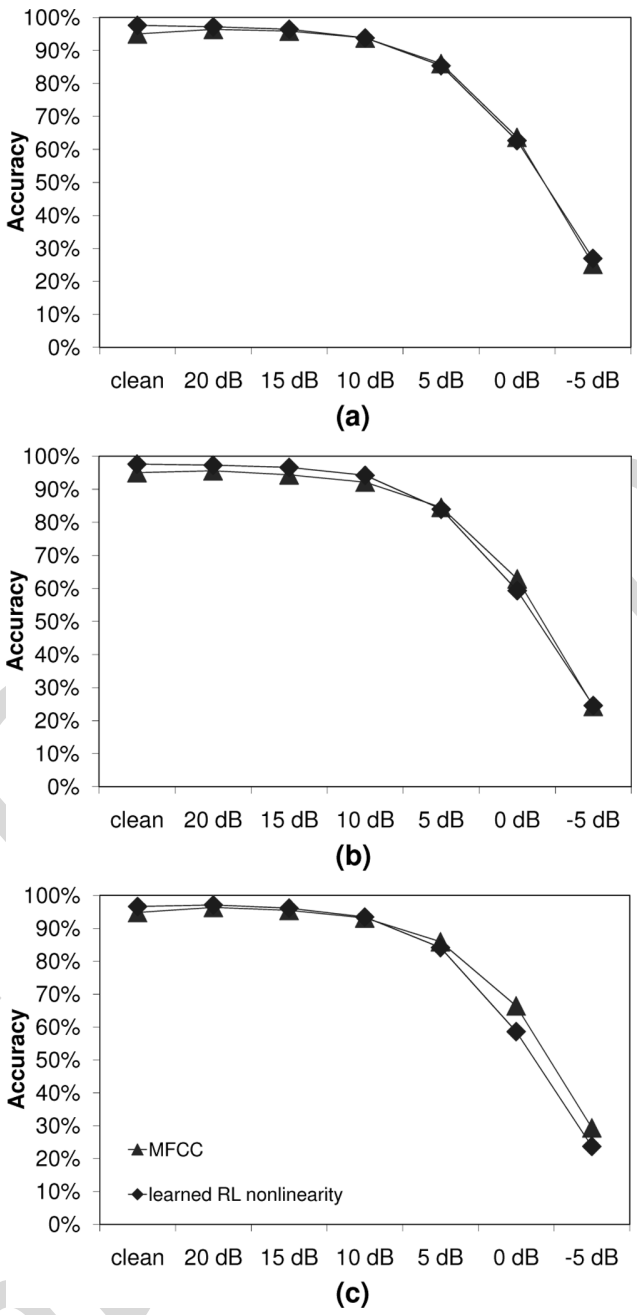


Fig. 12. Comparison of recognition accuracy in the presence of three sets of background noise for the AURORA 2 corpus using multi-style training. (a) Test set A. (b) test set B. (c) Test set C.

senones was used. These results confirm that recognition accuracy using the WSJ data follow trends that are similar to what has been previously described for the RM database. The optimization process provides an additional increase of 2 to 4 dB in effective SNR compared to the SNR obtained using the deterministic initial values of the parameters of the rate-level nonlinearity and an improvement of 3 to 5 dB compared to the baseline MFCC results.

E. Recognition Accuracy Using the AURORA 2 Database

Fig. 11 shows results obtained using the AURORA 2 database after training using clean speech. HMMs with 1000 tied states, each modeled by a mixture of eight Gaussians for MFCC coefficients, and 32 Gaussians for features obtained using the rate-level nonlinearity, were trained for recognition experiments. (It was found that the use of 32 Gaussians to characterize MFCC features provided better recognition accuracy for clean speech but higher error rates for the noisy conditions considered. 8 Gaussians per senone provided the best MFCC performance in the noisy evaluation conditions.) The feature extraction employed a 23-filter Mel filter bank covering the frequency range

of 64 Hz to 4000 Hz. The number of cepstral coefficients for best recognition accuracy was determined empirically to be 10 for MFCCs and 11 for the rate-level nonlinearity. The initial nonlinearity parameters were set to $w_0 = -0.110$, $w_1 = -0.521$, $\alpha = 0.05$ to account for the change of sampling rate from 16 kHz to 8 kHz.

The results of Fig. 11 indicate that recognition accuracy using the AURORA 2 database follows similar trends to what had been previously described for the RM and WSJ databases. The optimization process provides an additional 2 to 4 dB increase in effective SNR compared to the SNR obtained using the deterministic initials of rate-level nonlinearity and an improvement of 5 to 7 dB compared to the baseline MFCC results.

Fig. 12 shows results obtained using the AURORA 2 corpus with multi-condition training. The use of the recognition system with the learned sigmoidal rate-level nonlinearity does not appear to provide much benefit compared to baseline MFCC processing when multi-condition is employed.

V. Summary and Conclusion

In a previous study [24], we found that the sigmoidal rate-level nonlinearity that is a part of most models of the physiological transformation of sound to its neural representation contributes the most to robustness in speech recognition, especially when there is a mismatch of training and testing environments. In this paper we model this nonlinearity by a set of frequency-dependent logistic functions, and we develop an automated procedure for learning the optimal values of the parameters of these functions from training data using an objective function based on maximum mutual information. This function is coupled with a complementary function that models the observed psychoacoustical equal-loudness contour, and the two functions are inserted into the chain of operations that constitutes MFCC processing.

The process of learning the optimal parameters of the rate-level nonlinearities is sped up very substantially through the use of lattice information generated from the speech decoder to prune out unlikely state sequences, and through the use of conjugate gradient descent that reduces the total number of iterations required to achieve convergence. Together these improvements speed up the learning process by a factor of approximately 25.

Using equal-loudness compensation and the learned sigmoidal rate-level nonlinearity, we observed a typical improvement of approximately 5 to 7 dB in effective SNR compared to baseline MFCC processing at an SNR of 10 dB, and an improvement of 2 to 3 dB in effective SNR compared to a basic sigmoidal nonlinearity without the learning procedures described in this paper, when the system is trained on clean speech. These improvements in performance disappear when the system is trained and tested in multi-style fashion.

The algorithm described in Section III assumes that each of the phoneme classes is modeled by a single Gaussian. It is natural to hypothesize that having more detailed distributions, e.g., mixtures of Gaussians, could result in better learned sigmoidal parameters. The modifications required in the algorithm to deal with mixtures of Gaussians are minimal as only the *a posteriori* probabilities of individual Gaussians in the mixture need be considered. However, the increase in computation is significant, and in our experiments the benefit obtained from scaling up from single Gaussians to mixtures of Gaussians was marginal and did not justify the large increase in computation that it entailed.

Another natural question that arises is that of what happens if the sigmoidal parameters are learned from only clean speech. We note that the purpose of learning the parameters of the nonlinearity in the fashion described in this paper is to reduce the differences between features computed from clean speech and those obtained from noisy speech. Therefore, training sigmoidal parameters from noisy speech is an integral aspect of the algorithm. Nevertheless we did conduct an experiment where we learned the sigmoidal parameters from only clean speech. Not surprisingly, while performance on clean speech improved, it did not improve performance on noisy speech. This was to be expected: since the optimal nonlinearity is learned from data, it cannot possibly become robust to noise without being exposed to noise.

In a related study [30], we demonstrated that an additional improvement in recognition accuracy can be obtained by combining the learned rate-level nonlinearity with post-processing techniques such as modulation filtering of the cepstral-like coefficients that are derived from the processing described here. Nevertheless, we believe that further improvements can be obtained by fully integrating the benefits of all of these methods into a single algorithm that provides a joint optimization over both the parameters characterizing the nonlinearity parameters and the parameters that determine the modulation filter.

Appendix
Derivation of the Derivative Update Equation

With the assumption that the prior probabilities of each class are equal and that the observation probability $P(\mathbf{s}|C)$ is a single Gaussian, the feature vector $\mathbf{s}$ of the classifier can be computed from the input vector $\mathbf{x}$ using the rate-level nonlinearity and the DCT transformation

$$s_{u,t}[k] = \beta[k] \sum_{n=1}^{N} x_{u,t}[n] cos \frac{\pi(2n-1)(k-1)}{2N}$$
$$k = 1, \ldots, K, \; with$$
$$\beta[k] = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } k = 1 \\ \sqrt{\frac{2}{N}}, & \text{otherwise} \end{cases} \tag{11}$$

where $x_{u,t}[n]$ is given in (1) and K is the number of MFCC coefficients. (We used a value of $K = 13$ in the present paper.) The overall accumulated posterior probability can be written as

$$P = \prod_u \prod_t \frac{N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_C, \boldsymbol{\sigma}_C)}{\sum_{C'} N(\mathbf{s}^c_{u,t}|\boldsymbol{\mu}_{C'}, \boldsymbol{\sigma}_{C'})}. \tag{12}$$

In the above equations, u denotes the utterance index, t denotes the time index in each utterance, and $\mathbf{s}^c_{u,t}$ denotes the fea-

tures of the incoming utterance after cepstral mean substraction (CMS) has been applied, that $\mathbf{s}^c_{u,t} = \mathbf{s}_{u,t} - (1/T_u)\sum_{t=1}^{T_u}\mathbf{s}_{u,t}$:

$$\begin{aligned} &Objective = \max \log P \\ &= \max \sum_u \sum_t \left[-\frac{1}{2}\sum_{k=1}^{K}\left[\log\sigma_C[k]^2 + \frac{\|s^c_{u,t}[k]-\mu_C[k]\|^2}{\sigma_C[k]^2}\right] \right. \\ &\left. - \log \sum_{C'}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[k]^2}} e^{-(\|s^c_{u,t}[k]-\mu_{C'}[k]\|^2/2\sigma_{C'}[k]^2)}\right]. \end{aligned} \tag{13}$$

Taking the derivative with respect to $\mathbf{F} = \{\boldsymbol{\alpha}, w_0, w_1\}$ we obtain

$$\begin{aligned} &\frac{\partial \log P}{\partial \mathbf{F}} \\ &= \sum_u\sum_t\left[-\frac{1}{2}\sum_{k=1}^{K}\left[\frac{\partial \log\sigma_C[k]^2}{\partial\mathbf{F}} + \frac{\partial}{\partial\mathbf{F}}\frac{\|s^c_{u,t}[k]-\mu_C[k]\|^2}{\sigma_C[k]^2}\right]\right. \\ &\left. - \frac{\partial}{\partial\mathbf{F}}\log\sum_{C'}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[k]^2}} e^{-(\|s^c_{u,t}[k]-\mu_{C'}[k]\|^2/2\sigma_{C'}[k]^2)}\right] \end{aligned} \tag{14}$$

$$\begin{aligned} &\frac{\partial\{-\frac{1}{2}\log\sigma_C[k]^2\}}{\partial\mathbf{F}} = -\frac{1}{2\sigma_C[k]^2}\frac{\partial\sigma_C[k]^2}{\partial\mathbf{F}}, \\ &\frac{\partial}{\partial\mathbf{F}}\left[-\frac{1}{2}\frac{\|s^c_{u,t}[k]-\mu_C[k]\|^2}{\sigma_C[k]^2}\right] = \\ &-\frac{1}{2\sigma_C[k]^4}\left[2(s^c_{u,t}[k]-\mu_C[k])\left(\frac{\partial s^c_{u,t}[k]}{\partial\mathbf{F}} - \frac{\partial\mu_C[k]}{\partial\mathbf{F}}\right)\sigma_C[k]^2\right. \\ &\left. - \|s^c_{u,t}[k]-\mu_C[k]\|^2\frac{\partial\sigma_C[k]^2}{\partial\mathbf{F}}\right], \\ &\frac{\partial}{\partial\mathbf{F}}\left[\log\sum_{C'}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[k]^2}}e^{(-\|s^c_{u,t}[k]-\mu_{C'}[k]\|^2/2\sigma_{C'}[k]^2)}\right] \\ &= \frac{\frac{\partial}{\partial\mathbf{F}}\left[\sum_{C''}\prod_{k=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C''}[k]^2}}e^{-(\|s^c_{u,t}[k]-\mu_{C''}[k]\|^2/2\sigma_{C''}[k]^2)}\right]}{\sum_{C'}\prod_{l=1}^{K}\frac{1}{\sqrt{2\pi\sigma_{C'}[l]^2}}e^{-(\|s^c_{u,t}[l]-\mu_{C'}[l]\|^2/2\sigma_{C'}[l]^2)}} \\ &= \sum_{C''}\left[\sum_{k=1}^{K}\left[-\frac{1}{2\sigma_{C''}[k]^2}\frac{\partial\sigma_{C''}[k]^2}{\partial\mathbf{F}}\right.\right. \\ &\left.\left. - \frac{1}{2}\frac{\partial}{\partial\mathbf{F}}\left(\frac{\|s^c_{u,t}[k]-\mu_{C''}[k]\|^2}{\sigma_{C''}[k]^2}\right)\right]\right] \\ &\cdot\frac{\prod_{m=1}^{K}\frac{1}{\sigma_{C''}[m]}e^{-(\|s^c_{u,t}[m]-\mu_{C''}[m]\|^2/2\sigma_{C''}[m]^2)}}{\sum_{C'}\prod_{l=1}^{K}\frac{1}{\sigma_{C'}[l]}e^{-(\|sal^c_{u,t}[l]-\mu_{C'}[l]\|^2/2\sigma_{C'}[l]^2)}}. \end{aligned} \tag{15}$$

The model parameters $\boldsymbol{\mu}_C$ and $\boldsymbol{\sigma}_C$ were obtained in the maximum likelihood sense in the same fashion as in training the speech recognizer [(8) with mean substraction]:

$$\begin{aligned} \sigma_C[k] &= \frac{1}{\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C)}\sum_u\sum_t^{T_u} I(\mathbf{s}_{u,t}\in C) \\ &\cdot\left(s_{u,t}[k] - \frac{1}{T_u}\sum_{t=1}^{T_u}s_{u,t}[k] - \mu_C[k]\right)^2, \\ \mu_C[k] &= \frac{1}{\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C)}\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C) \\ &\cdot\left(s_{u,t}[k] - \frac{1}{T_u}\sum_{t=1}^{T_u}s_{u,t}[k]\right). \end{aligned} \tag{16}$$

The partial derivative of mean and variance of each class and feature vector $s^c_{u,t}[k]$ over $\mathbf{F}$ can be written as

$$\begin{aligned} \frac{\partial\sigma_C[k]^2}{\partial\mathbf{F}} &= \frac{2}{N_C}\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C) \\ &\times\left(\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} - \frac{1}{T_u}\sum_{t=1}^{T_u}\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} - \frac{\partial\mu_C[k]}{\partial\mathbf{F}}\right) \\ &\times\left(s_{u,t}[k] - \frac{1}{T_u}\sum_{t=1}^{T_u}s_{u,t}[k] - \mu_C[k]\right), \\ \frac{\partial\mu_C[k]}{\partial\mathbf{F}} &= \frac{1}{N_C}\sum_u\sum_{t=1}^{T_u} I(\mathbf{s}_{u,t}\in C)\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} \\ &- \frac{1}{T_u}\sum_u\sum_{t=1}^{T_u}\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}}, \\ \frac{\partial s^c_{u,t}[k]}{\partial\mathbf{F}} &= \frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} - \frac{1}{T_u}\sum_{t=1}^{T_u}\frac{\partial s_{u,t}[k]}{\partial\mathbf{F}} \end{aligned} \tag{17}$$

where N_C is the number of frames in class C and T_u is the number of frames in each utterance. We note that the above feature computation incorporates CMS. Furthermore, the mean of the sentence we are subtracting is not a constant developed during training, but rather it is taken as an average over the frames of the corresponding utterance. This occurs when we take the directive with respect to the nonlinearity parameters, which occurs in the second term of the last line of (17). In addition,

$$\begin{aligned} &\frac{\partial s_{u,t}[k]}{\partial\alpha[o]} = \frac{\partial}{\partial\alpha[o]} \\ &\cdot\left(\beta[k]\sum_{n=1}^{N}\frac{\alpha[n]}{1+e^{w_1[n]\cdot y_{u,t}[n]+w_0[n]}}cos\frac{\pi(2n-1)(k-1)}{2N}\right) \\ &= \beta[k]\frac{1}{1+e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}}cos\frac{\pi(2o-1)(k-1)}{2N} \\ &\frac{\partial s_{u,t}[k]}{\partial w_0[o]} \end{aligned}$$

$$= -\beta[k]\frac{e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}}{\left(1+e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}\right)^2}cos\frac{\pi(2o-1)(k-1)}{2N}$$

$$\frac{\partial s_{u,t}[k]}{\partial w_1[o]}$$

$$= -\beta[k]\frac{y_{u,t}[o]\cdot e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}}{\left(1+e^{w_1[o]\cdot y_{u,t}[o]+w_0[o]}\right)^2}cos\frac{\pi(2o-1)(k-1)}{2N}. \quad (18)$$

REFERENCES

[1] E. D. Young and M. B. Sachs, "Representation of steady state vowels in the temporal aspects of the discharge patterns of populations of auditory-nerve fibers," *J. Acoustic. Soc. Amer.*, vol. 66, pp. 1381–1403, 1979.

[2] N. F. Viemeister, "Auditory intensity discrimination at high frequencies in the presence of noise," *Science*, vol. 221, pp. 1206–1208, 1983.

[3] R. L. Winslow and M. B. Sachs, "Single tone intensity discrimination based on auditory-nerve rate responses in backgrounds of quiet, noise and stimulation of the crossed olivocochlcar bundle," *Hear. Res.*, vol. 35, pp. 165–190, 1988.

[4] I. M. Winter and A. R. Palmer, "Intensity coding in low-frequency auditory-nerve fibers of the guinea pig," *J. Acoust. Soc. Amer.*, vol. 90, no. 4, pp. 1958–1967, 1991.

[5] J. Volkmann, S. S. Stevens, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. Acoust. Soc. Amer.*, vol. 8, no. 3, pp. 208–209, 1937.

[6] E. Zwicker, "Subdivision of the audible frequency range into critical bands," *J. Acoust. Soc. Amer.*, vol. 33, no. 2, pp. 248–248, 1961.

[7] M. B. Sachs and P. J. Abbas, "Rate versus level functions for auditory-nerve fibers in cats: Tone-burst stimuli," *J. Acoust. Soc. Amer.*, vol. 56, pp. 1835–1847, 1974.

[8] M. B. Sachs, R. L. Winslow, and B. H. A. Sokolowski, "A computational model for rate-level functions from cat auditory-nerve fibers," *Hear. Res.*, vol. 41, pp. 61–70, 1989.

[9] M. B. Sachs and N. Y. Kiang, "Two-tone inhibition in auditory-nerve fibers," *J. Acoust. Soc. Amer.*, vol. 43, pp. 1120–1128, 1968.

[10] P. J. Abbas and M. B. Sachs, "Two-tone suppression in auditory-nerve fibers: Extension of a stimulus-response relationship," *J. Acoust. Soc. Amer.*, vol. 59, pp. 112–122, 1976.

[11] L. L. Elliott, "Changes in the simultaneous masked threshold of brief tones," *J. Acoust. Soc. Amer.*, vol. 38, pp. 738–746, 1965.

[12] E. Zwicker, "Temporal effects in simultaneous masking and loudness," *J. Acoust. Soc. Amer.*, vol. 38, pp. 132–141, 1965.

[13] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[14] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, pp. 1738–1752, 1990.

[15] O. Ghitza, "Auditory nerve representation as a front-end for speech recognition in a noisy environment," *Comput. Speech Lang.*, vol. 1, no. 2, pp. 109–131, 1986.

[16] S. Seneff, "A joint synchrony/mean-rate model of auditory speech processing," *J. Phonetics*, vol. 15, pp. 55–76, 1988.

[17] A. Biem, S. Katagiri, E. McDermott, and B.-H. Juang, "An application of discriminative feature extraction to filter-bank-based speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 9, no. 2, pp. 96–110, Feb. 2001.

[18] T. Kinnunen, "Design a speaker-discriminative adaptive filter bank for speaker recognition," in *Proc. Int Conf. Spoken Lang. Process.*, Denver, CO, Sep. 2002.

[19] T. Kobayashi and S. Imai, "Spectral analysis using generalized cepstrum," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 5, pp. 1087–1089, Oct. 1984.

[20] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis—A unified approach to speech spectral estimation," in *Proc. Int. Conf. Spoken Lang. Process.*, 1994.

[21] R. Sarikaya and J. H. L. Hansen, "Analysis of the root cepstrum for acoustic modeling and fast decoding in speech recognition," *Proc. Eurospeech*, 2001.

[22] C. K. S. Chatterjee and W. B. Kleijn, "Auditory model based optimization of MFCCs inproves automatic speech recognition performance," in *Proc. Interspeech*, Brighton, U.K., 2009.

[23] S. Chatterjee and W. B. Kleijn, "Auditory model based modified MFCC features," in *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, Dallas, TX, 2010, pp. 4590–4593.

[24] Y.-H. Chiu and R. M. Stern, "Analysis of physiologically-motivated signal processing for robust speech recognition," in *Proc. Interspeech*, Brisbane, Australia, Sep. 2008.

[25] M. C. Liberman, "Auditory nerve response from cats raised in a low-noise chamber," *J. Acoust. Soc. Amer.*, vol. 63, pp. 442–455, 1978.

[26] R. F. Lyon, "A computational model of filtering, detection and compression in the cochlea," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Paris, France, May 1982, pp. 1282–1285.

[27] M. G. Heinz, X. Zhang, I. C. Bruce, and L. H. Carney, "Auditory nerve model for predicting performance limits of normal and impaired listeners," *Acoust. Res. Lett. Online*, vol. 2, no. 3, pp. 91–96, 2001.

[28] Y.-H. Chiu, B. Raj, and R. Stern, "Towards fusion of feature extraction and acoustic model training: A top down process for robust speech recognition," in *Proc. Interspeech*, Brighton, U.K., 2009.

[29] Y.-H. Chiu and R. M. Stern, "Learning-based auditory encoding for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, Apr. 2010, pp. 4278–4281.

[30] Y.-H. Chiu and R. M. Stern, "Minimum variance modulation filter for robust speech recognition," in *Proc. IEEE Conf. Acoustics, Speech, Signal Process.*, Taipei, Taiwan, 2009, pp. 3917–3920.

[31] E. Terhardt, "Calculating virtual pitch," *Hear. Res.*, vol. 1, pp. 155–182, 1979.

[32] A. S. Bregman, *Auditory Scene Analysis*. Cambridge, MA: MIT Press, 1990.

[33] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "On a model-robust training method for speech recognition," in *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, Tokyo, Japan, Apr. 1986.

[34] M. A. P. A. Nadas and D. Nahamoo, "On a model-robust training method for speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1432–1436, Sep. 1988.

[35] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, pp. 525–533, 1993.

[36] J. R. Shewchuk, An introduction to the conjugate gradient method without the agonizing pain Comput. Sci. Dept., Carnegie Mellon Univ. , 1994, Tech. Rep. CS-94-125.

[37] A. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones, The NOISEX-92 study on the effect of additive noise on automatic speech recognition DRA Speech Research Unit, Malvern, U.K., 1992, Tech. Rep..

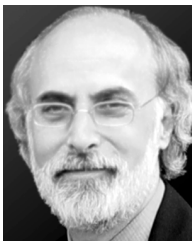
[38] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, pp. 247–251, 1993.

Yu-Hsiang Bosco Chiu (M'10) received the B.S. and M.S. degrees from the Electrical Engineering Department, National Tsing Hua University, Hsinchu, Taiwan, in 2001 and 2003, respectively, and the Ph.D. degree from the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA. in 2010.

His research is in speech recognition and language understanding, where he has focused on the development of automatic learning algorithms for enhancing speech recognition performance under adverse conditions. He is interested in computational perception algorithms that are loosely motivated by physiological principles and that are optimized for best recognition performance.

Bhiksha Raj (M'10) received the Ph.D. degree from Carnegie Mellon University (CMU), Pittsburgh, PA. in 2000.

From 2000 to 2001, he was at Compaq's Cambridge Research Labs, Boston, and from 2001 to 2008 he headed the speech research effort at Mitsubishi Electric Research Labs. Since the fall of 2008, he has been an Associate Professor at the Language Technologies Institute, Carnegie Mellon University, as well as an Associate Professor by Courtesy in CMU's Department of Electrical and Computer Engineering. He has conducted research in a variety of areas including noise robust speech recognition, likelihood-maximizing beamforming, data visualization, and latent-variable spectral decompositions for signal separation. He has also been a major contributor to the Sphinx suite of open-source systems, and he served as the main architect of Sphinx 4. At Mitsubishi, he was primarily responsible for the invention and development of techniques for voice-based search, many of which were highly successful. He holds several patents (and patent applications) in speech recognition, voice search and denoising, and he is the author of over 100 articles in refereed conferences, journals, and books.



Richard M. Stern (M'76) received the B.S. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1970, the M.S. degree from the University of California, Berkeley, in 1972, and the Ph.D. degree from MIT in 1977, all in electrical engineering.

He has been on the faculty of Carnegie Mellon University, Pittsburgh, PA, since 1977, where he is currently a Professor in the Electrical and Computer Engineering, Computer Science, and Biomedical Engineering Departments, and the Language Technologies Institute. Much of his current research is in spoken language systems, where he is particularly concerned with the development of techniques with which automatic speech recognition can be made more robust with respect to changes in environment and acoustical ambience. He has also developed sentence parsing and speaker adaptation algorithms for earlier CMU speech systems. In addition to his work in speech recognition, he also maintains an active research program in psychoacoustics, where he is best known for theoretical work in binaural perception.

Dr. Stern is a Fellow of the Acoustical Society of America and the International Speech Communication Association (ISCA), the 2008–2009 ISCA Distinguished Lecturer, a recipient of the Allen Newell Award for Research Excellence in 1992, and he served as General Chair of Interspeech 2006. He is also a member of the Audio Engineering Society.