VARIATIONAL BAYESIAN LEARNING OF SPEECH GMMS FOR FEATURE ENHANCEMENT BASED ON ALGONQUIN

Svein G. Pettersen, Magne H. Johnsen*

Norwegian University of Science and Technology
Department of Electronics and Telecommunications
NO-7491 Trondheim, Norway
sveingun@iet.ntnu.no, mhj@iet.ntnu.no

Christian Wellekens

Institut Eurecom
2229 Route des Cretes
06904 Sophia Antipolis, France
Christian.Wellekens@eurecom.fr

ABSTRACT

Many feature enhancement methods make use of probabilistic models of speech and noise in order to improve performance of speech recognizers in the presence of background noise. The traditional approach for training such models is maximum likelihood estimation. This paper investigates the novel application of variational Bayesian learning for front-end models under the Algonquin denoising framework. Compared to maximum likelihood training, it is shown that variational Bayesian learning has advantages both in terms of increased robustness with respect to choice of model complexity, as well as increased performance.

Index Terms— Speech recognition, Speech enhancement, Robustness, Variational methods

1. INTRODUCTION

The performance of speech recognizers can drop significantly in the presence of background noise. If the recognition models have been trained in clean conditions, the problem becomes even more severe due to the resulting mismatch between training and test conditions. To improve robustness in such situations, there are two basic categories of compensation approaches: feature-based compensation and model-based compensation. Feature-based compensation schemes aim to estimate clean speech from noise-corrupted speech based on a noise-model or knowledge about how the noise changes the signal statistics, while model-based compensation schemes adjust the system parameters in order to obtain a model better suited for recognition in the noisy environment.

The main advantage of feature-based approaches compared to model-based approaches is that they are computationally less demanding. For the feature-based methods there has been increasing interest in taking advantage of probabilistic models of speech and noise [1, 2, 3, 4]. Such front-end models are normally chosen to be much simpler than the recognizer models in order to retain the advantage of computational simplicity. A common choice is the Gaussian mixture model (GMM).

Training of such probabilistic models is also an active field of research. Approximate Bayesian learning has been made possible during the past few years through the use of variational methods [5]. Variational Bayesian (VB) training offers several advantages over traditional maximum likelihood (ML) training. Examples of previous applications of VB learning to speech recognition are training of GMMs for recognition of confusable phones [6] and HMM model-selection and training for large vocabulary speech recognition [7]. The novelty of this paper is the application of VB training to front-end models for speech denoising using the Algonquin framework [3]. Compared to ML training, we show that the VB approach gives increased robustness w.r.t. choice of model complexity, as well as increased performance.

This paper starts by briefly reviewing variational Bayesian learning in section 2, before describing VB training of the GMM in section 3. Then, we review the Algonquin algorithm for denoising in section 4. Section 5 describes the application of VB trained models to feature enhancement and our motivations for doing so. Experiments and results are presented in section 6 before the conclusion in section 7.

2. VARIATIONAL BAYESIAN LEARNING

In [5] Attias proposed a variational approach for Bayesian learning of graphical models. Let $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ denote the observed data, $S = \{\mathbf{s}_1, \cdots, \mathbf{s}_N\}$ denote hidden variables and Θ denote the parameters. For a given model structure m the goal is to compute the parameter posterior $p(\Theta|X, m)$. In addition, for the purpose of model selection, the posterior of model structures $p(m|X)$ is of interest.

To make Bayesian computations tractable, the key point is to approximate the joint posterior $p(S, \Theta|X)$ by a variational posterior $q(S, \Theta|X)$ which is restricted to a factorized form as

$$q(S, \Theta|X) = q(S|X)q(\Theta|X). \tag{1}$$

Note that q should always be understood as conditioned on X, although it is common not to write this explicitly. We will also follow this convention. It is now possible to reformulate the problem of computing the posterior as an optimization problem, where the cost function $\mathcal{F}_m$ is defined by

$$\mathcal{F}_m = \int q(S)q(\Theta) \log \frac{p(X, S, \Theta)}{q(S)q(\Theta)} d\Theta dS. \tag{2}$$

This quantity is also often referred to as *free energy*. It follows from Jensen's inequality that $\mathcal{F}_m$ is bounded from above by the marginal log likelihood, i.e.

$$\mathcal{F}_m \leq \log p(X|m). \tag{3}$$

The objective function can also be written as

$$\mathcal{F}_m = E_{S,\Theta}\left[\log \frac{p(X, S|\Theta)}{q(S)}\right] - KL[q(\Theta)||p(\Theta)], \tag{4}$$

*The work was done while S.G. Pettersen was visiting Institut Eurecom.

where $E_{S,\Theta}[\cdot]$ denotes the expectation w.r.t. $q(S,\Theta)$ and KL denotes the Kullback-Leibler distance. While the first term corresponds to the averaged likelihood, the second term can be interpreted as a penalty term for more complex models. As we increase the number of parameters in order to increase the average likelihood, the KL distance will also increase and thus reduce the total value of $\mathcal{F}_m$. Assuming equal prior probabilities for all model structures m, the model with the highest value of $\mathcal{F}_m$ corresponds to the model with the highest posterior probability.

For optimization of the objective function, an EM-like algorithm is used. The E-step consists of computing the variational posterior over hidden variables as

$$q(S) \propto \exp\left\{E_\Theta[\log p(X,S|\Theta)]\right\}. \tag{5}$$

The M-step is then to compute the variational parameter posterior as

$$q(\Theta) \propto \exp\{E_S[\log p(X,S|\Theta)]\}p(\Theta). \tag{6}$$

3. VB LEARNING FOR GMM

The application of VB learning as described in section 2 for the GMM was also presented in [5].

The GMM has the form

$$p(\mathbf{x}_n|\Theta,m) = \sum_{s=1}^{m} p(\mathbf{x}_n|s_n=s,\Theta)p(s_n=s|\Theta) \tag{7}$$

where s_n denotes the hidden component that generated observation $\mathbf{x}_n$. In this case, the model structure is simply the number of components, denoted m. Each component has a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_s,\boldsymbol{\Gamma}_s)$, where $\boldsymbol{\mu}_s$ is the mean vector, and $\boldsymbol{\Gamma}_s$ is the precision matrix. The mixture weight for component s is denoted by π_s, i.e. $p(s_n=s|\Theta)=\pi_s$.

It is useful to choose prior densities from conjugate families, since the posterior densities will then belong to the same families as the priors. As a consequence, the VB learning simply amounts to updating the hyperparameters of the posteriors. Thus, the following conjugate priors are defined for the parameters Θ.

$$p(\{\pi_s\}) = \mathcal{D}(\lambda^0) \tag{8}$$

$$p(\boldsymbol{\mu}_s|\boldsymbol{\Gamma}_s) = \mathcal{N}(\boldsymbol{\rho}^0,\beta^0\boldsymbol{\Gamma}_s) \tag{9}$$

$$p(\boldsymbol{\Gamma}_s) = \mathcal{W}(\nu^0,\Phi^0) \tag{10}$$

Here $\mathcal{D}$ and $\mathcal{W}$ denote Dirichlet and Wishart densities respectively.

Define $\gamma_s^n = q(s_n=s|\mathbf{x}_n)$. The objective of the E-step is to compute γ_s^n, and this can then be done as follows:

$$\gamma_s^n \propto \tilde{\pi}_s\tilde{\Gamma}_s^{1/2}e^{-(\mathbf{x}_n-\boldsymbol{\rho}_s)^T\bar{\boldsymbol{\Gamma}}_s(\mathbf{x}_n-\boldsymbol{\rho}_s)/2}e^{-d/2\beta_s}, \tag{11}$$

where

$$\log\tilde{\pi}_s = E_\Theta[\log\pi_s] = \psi(\lambda_s) - \psi\left(\sum_{s'}\lambda_{s'}\right) \tag{12}$$

$$\begin{aligned}\log\tilde{\Gamma}_s &= E_\Theta[\log|\boldsymbol{\Gamma}_s|]\\ &= \sum_{i=1}^{d}\psi\left(\frac{\nu_s+1-i}{2}\right) - \log|\boldsymbol{\Phi}_s| + d\log 2\end{aligned} \tag{13}$$

$$\bar{\boldsymbol{\Gamma}}_s = E_\Theta[\boldsymbol{\Gamma}_s] = \nu_s\boldsymbol{\Phi}_s^{-1}. \tag{14}$$

In the above equations, ψ denotes the digamma function. The normalization constant of γ_s^n can be found using the constraint that $\sum_{s=1}^m\gamma_s^n=1$.

The M-step can be divided into two stages. In the first stage, which is the same as in the ordinary EM algorithm, the following quantities are computed.

$$\bar{\pi}_s = \frac{1}{N}\sum_{n=1}^{N}\gamma_s^n \tag{15}$$

$$\bar{\boldsymbol{\mu}}_s = \frac{1}{\bar{N}_s}\sum_{n=1}^{N}\gamma_s^n\mathbf{x}_n \tag{16}$$

$$\bar{\boldsymbol{\Sigma}}_s = \frac{1}{\bar{N}_s}\sum_{n=1}^{N}\gamma_s^n\mathbf{C}_s^n \tag{17}$$

Here, $\mathbf{C}_s^n = (\mathbf{x}_n-\bar{\boldsymbol{\mu}}_s)(\mathbf{x}_n-\bar{\boldsymbol{\mu}}_s)^T$, and $\bar{N}_s = N\bar{\pi}_s$. The hyperparameters of the posteriors are then updated in the second stage.

$$\lambda_s = \bar{N}_s+\lambda^0, \quad \nu_s = \bar{N}_s+\nu^0, \quad \beta_s = \bar{N}_s+\beta^0 \tag{18}$$

$$\boldsymbol{\rho}_s = \frac{\bar{N}_s\bar{\boldsymbol{\mu}}_s+\beta^0\boldsymbol{\rho}^0}{\bar{N}_s+\beta^0} \tag{19}$$

$$\boldsymbol{\Phi}_s = \bar{N}_s\bar{\boldsymbol{\Sigma}}_s + \frac{\bar{N}_s\beta^0}{\bar{N}_s+\beta^0}(\bar{\boldsymbol{\mu}}_s-\boldsymbol{\rho}^0)(\bar{\boldsymbol{\mu}}_s-\boldsymbol{\rho}^0)^T + \boldsymbol{\Phi}^0 \tag{20}$$

Since posteriors are computed instead of parameters, the predictive density is used for unseen data. In this density the parameters Θ are integrated out. This gives us a mixture of multivariate t-distributions on the form

$$p(\mathbf{x}|X) = \sum_{s=1}^{m}\bar{\pi}_s t_{\omega_s}(\mathbf{x}|\boldsymbol{\rho}_s,\boldsymbol{\Lambda}_s). \tag{21}$$

For component s, the degrees of freedom are $\omega_s=\nu_s+1-d$, the mean is $\boldsymbol{\rho}_s$ and the covariance is $\boldsymbol{\Lambda}_s = ((\beta_s+1)/\beta_s\omega_s)\boldsymbol{\Phi}_s$. The mixture weight is given by $\bar{\pi}_s = \lambda_s/\sum_{s'}\lambda_{s'}$.

4. FEATURE ENHANCEMENT BASED ON ALGONQUIN

Algonquin [3, 8] is a feature cleaning method that typically operates in the log-spectrum domain. The method takes advantage of probabilistic models of speech and noise in order to remove additive noise. (Note that the method can also handle distortion from the channel, but this will not be considered here.)

Let $\mathbf{x}$, $\mathbf{n}$ and $\mathbf{y}$ denote clean speech, noise and noisy speech vectors respectively. Given GMM priors $p(\mathbf{x})$ and $p(\mathbf{n})$ for speech and noise, the Algonquin method uses a variational algorithm to find an approximation of the posterior $p(\mathbf{x}|\mathbf{y})$. This variational posterior is denoted as $q_\mathbf{y}(\mathbf{x})$. Given $\mathbf{x}$ and $\mathbf{n}$, the distribution of $\mathbf{y}$ is modeled as

$$p(\mathbf{y}|\mathbf{x},\mathbf{n}) = \mathcal{N}(\mathbf{y};\mathbf{x}+\log(1+\exp(\mathbf{n}-\mathbf{x})),\boldsymbol{\Psi}) \tag{22}$$

where $\boldsymbol{\Psi}$ is the covariance matrix. Thus, the joint distribution between $\mathbf{x}$, $\mathbf{n}$, speech GMM component s^x and noise GMM component s^n is given by

$$\begin{aligned}p(\mathbf{y},\mathbf{x},\mathbf{n},s^x,s^n) &= p(\mathbf{y}|\mathbf{x},\mathbf{n})p(s^x)p(\mathbf{x}|s^x)p(s^n)p(\mathbf{n}|s^n)\\ &= \mathcal{N}(\mathbf{y};\mathbf{g}(\mathbf{x},\mathbf{n}),\boldsymbol{\Psi})\pi_{s^x}^x\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_x^{s^x},\boldsymbol{\Sigma}_x^{s^x})\\ &\quad\cdot\pi_{s^n}^n\mathcal{N}(\mathbf{n};\boldsymbol{\mu}_n^{s^n},\boldsymbol{\Sigma}_n^{s^n})\end{aligned} \tag{23}$$

where $\mathbf{g}(\mathbf{x},\mathbf{n}) = \mathbf{x}+\log(1+\exp(\mathbf{n}-\mathbf{x}))$. Algonquin approximates the true posterior by a variational distribution which is also modeled as a GMM, i.e.

$$q_\mathbf{y}(\mathbf{x},\mathbf{n}) = \sum_{\{s^x,s^n\}}q(s^x,s^n)q_\mathbf{y}(\mathbf{x},\mathbf{n}|s^x,s^n). \tag{24}$$

	Set 1		Set 2	
m	ML	VB	ML	VB
10	70.22	71.11	69.48	69.82
14	70.65	69.88	**70.19**	71.23
18	70.56	71.05	69.42	70.22
22	**72.28**	71.29	69.30	69.30
26	71.72	72.24	68.44	69.97
30	71.08	71.35	—	70.80
34	69.70	71.97	—	70.95
38	—	**72.74**	—	**71.26**
42	—	72.18	—	70.37
46	—	71.17	—	69.94
50	—	70.31	—	70.43

Table 1. Recognition performance (word accuracy) after denoising files containing subway noise at 5 dB, using models trained with two different training sets

	Set 3		Set 4	
m	ML	VB	ML	VB
10	69.48	70.53	70.10	70.68
14	69.54	69.20	67.64	69.88
18	**69.79**	70.22	**70.28**	70.86
22	68.93	69.63	69.97	70.92
26	69.17	71.02	67.88	70.37
30	68.38	71.02	69.39	71.85
34	—	70.80	—	**72.24**
38	—	**71.94**	—	71.72
42	—	71.02	—	70.74
46	—	71.57	—	70.62
50	—	71.05	—	71.85

Table 2. Recognition performance (word accuracy) after denoising files containing subway noise at 5 dB, using models trained with two different training sets

The parameters of this distribution are found by maximizing the following objective function.

$$F = \sum_{\{s^x, s^n\}} \int q_{\mathbf{y}}(\mathbf{x}, \mathbf{n}, s^x, s^n) \cdot \log \frac{p(\mathbf{y}, \mathbf{x}, \mathbf{n}, s^x, s^n)}{q_{\mathbf{y}}(\mathbf{x}, \mathbf{n}, s^x, s^n)} d\mathbf{x} d\mathbf{n} \quad (25)$$

See [3, 8] for details on the equations for the posterior parameters. After having found the posterior parameters, the minimum mean square error (MMSE) estimate of the clean speech feature vector is found as

$$\hat{\mathbf{x}} = \int \mathbf{x} p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \approx \int \mathbf{x} \sum_s q_{\mathbf{y}}(s) q_{\mathbf{y}}(\mathbf{x}|s) d\mathbf{x}. \quad (26)$$

5. TRAINING THE FRONT-END USING VB

There are several advantages to the Bayesian learning approach compared to traditional ML training. When only a small amount of data is available, ML training suffers from overfitting problems if the chosen model structure is too complex. In addition, if a component is assigned very few observations during ML training, numerical problems often arise. Because of the regularization effects from the priors, the VB training has no such numerical problems. In addition, since the VB objective function contains a penalty term for complex models, the training has an ability to prune the trained model according to the amount of data available. Thus, even if the model structure is chosen too complex, the model will not have the same overfitting problems as ML. Moreover, the VB free energy can be used as a model selection criterion to choose the right model complexity.

We applied the algorithm described in section 3 to train the speech prior used by Algonquin. The noise prior was an ML-trained single mixture estimated from the first 20 frames of each file, which are assumed to consist only of noise. The result of the VB training is posteriors for the parameters of $p(\mathbf{x})$. Ideally, we should use the predictive distribution given by eq. (21) when running Algonquin. However, since Algonquin is based on the assumption that the mixture components are Gaussian, we approximated each of the multivariate t-distributions with the multivariate Gaussian that was closest w.r.t. KL-distance. Given a component s it can be shown that this is a Gaussian with mean and covariance equal to that of the multivariate t-distribution [9], i.e.

$$p(\mathbf{x}) = \sum_{s=1}^{m} \bar{\pi}_s \mathcal{N}(\mathbf{x}; \boldsymbol{\rho}_s, \boldsymbol{\Lambda}_s). \quad (27)$$

6. EXPERIMENTS AND RESULTS

The experiments in this study were performed on Aurora2, which consists of spoken English digits with artificially added noise [10]. In order to investigate the performance of the feature enhancement algorithm in a case where only a small amount of data was available for training the front-end models, we used four different training sets, each consisting of 50 randomly selected files. Then, we trained models with different numbers of mixture components using both ML and VB. The training was initialized using the k-means algorithm. As a test set we chose the subset of Aurora2 containing subway noise at 5 dB.

For the VB training, the prior was set as follows:

$$\lambda^0 = 1, \quad \beta^0 = 1, \quad \nu^0 = d \quad (28)$$

$$\boldsymbol{\rho}^0 = \mathbf{0}, \quad \boldsymbol{\Phi}^0 = 10\mathbf{I} \quad (29)$$

where d denotes the feature dimension and $\mathbf{I}$ denotes the $d \times d$ identity matrix. The scaling constant for the prior covariance matrix $\boldsymbol{\Phi}^0$ was determined through some preliminary experiments.

Using the four different training sets, we trained prior models ($p(\mathbf{x})$) for Algonquin using ML and VB. Then, for each prior model we denoised all the files containing subway noise at 5 dB, before running the recognizer on the resulting files. Algonquin was performed in the log-spectrum domain, using 23 dimensional log filter bank feature vectors. After cleaning, these vectors were transformed into the cepstrum domain, where the feature vectors were 39 dimensional MFCCs including delta and acceleration parameters, with C_0 as energy. Using these feature vectors, recognition was then performed with models trained in clean condition. The baseline recognition result, using no feature cleaning, was a word accuracy of 45.26%.

The recognition results for models with number of mixture components m varying from 10 to 50 can be seen in tables 1 and 2. The ML results stop at around 30 mixtures where numerical problems arose due to lack of training data. However, the results show that for VB the results keep improving after that point. In addition, for a given model size VB is slightly better than ML in most cases. Figure 1 shows the results averaged over all four training sets from 2 to 50 mixtures. For very few mixtures, the results are almost identical. This is as expected since ML has enough training data to obtain

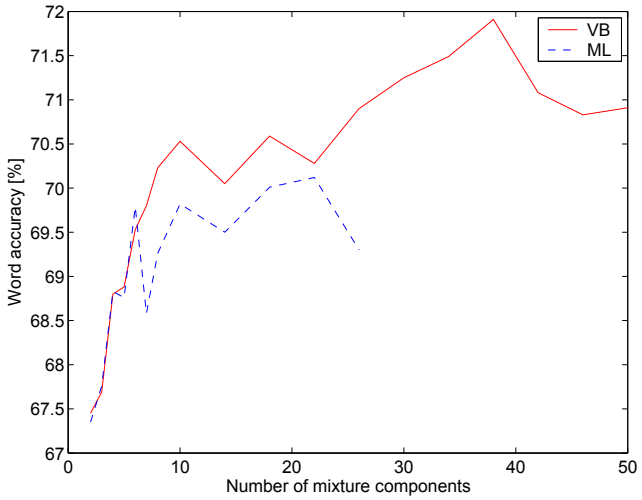

Fig. 1. Recognition performance after denoising files containing subway noise at 5 dB

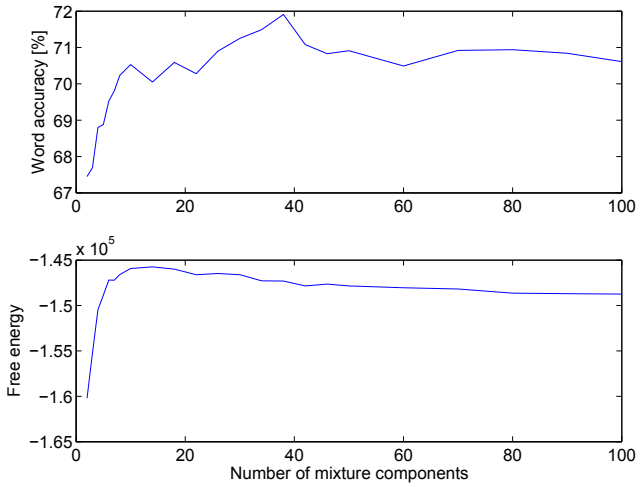

Fig. 2. Recognition performance for VB and free energy

robust parameter estimates. However, for model sizes greater than 7 VB performs better. This is probably because ML starts to suffer from overfitting problems. The VB learning, which has the advantage of the regularization effect of the priors, continues to improve until it reaches a peak at 38 mixtures.

In order to investigate what happens if the models size is increased even further, we ran experiments up to 100 mixtures, now in steps of 10. The value of the free energy was also examined. The top plot of figure 2 shows the recognition performance using VB, and the bottom plot shows the free energy. Both plots have been averaged over the four training sets. It can be seen that even if the model size is increased up to 100 mixtures, the performance remains at a reasonably high level. This is due to the self-pruning ability of the VB training. Since the free energy does not reflect the peak in performance at 38 mixtures, it cannot be used as a criterion for selecting the optimal model in this case. However, it still gives some useful information. In the range where the free energy is reasonably flat, the recognition performance is quite stable with a word accuracy of more than 70%.

7. CONCLUSION

In this paper we applied variational Bayesian learning to probabilistic models for feature enhancement. It was found that the Bayesian approach had advantages compared to traditional maximum likelihood training. The Bayesian approach avoided overfitting and numerical problems because of lack of training data as the model size was increased, and therefore resulted in improved recognition performance after enhancement.

8. ACKNOWLEDGMENTS

The authors thank Dr. Fabio Valente for helpful discussions. The work is done as a part of the BRAGE project, which is organized under the language technology programme KUNSTI and funded by the Norwegian Research Council.

9. REFERENCES

[1] H. Attias, J. C. Platt, A. Acero, and L. Deng, "Speech denoising and dereverberation using probabilistic models," in *Proc. NIPS*, 2000, pp. 758–764.

[2] L. Deng, A. Acero, M. Plumpe, and X.D. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proc. ICSLP*, 2000, pp. 806–809.

[3] B. J. Frey, T. T. Kristjansson, L. Deng, and A. Acero, "Algonquin - learning dynamic noise models from noisy speech for robust speech recognition," in *Proc. NIPS*, 2002.

[4] T.A. Myrvoll and S. Nakamura, "Online minimum mean square error filtering of noisy cepstral coefficients using a sequential em algorithm," in *Proc. ICSLP*, 2004, pp. 117–120.

[5] H. Attias, "A variational bayesian framework for graphical models," in *Proc. NIPS*, 2000, vol. 12, pp. 209–215.

[6] F. Valente and C.J. Wellekens, "Variational bayesian gmm for speech recognition," in *Proc. Eurospeech*, 2003, pp. 441–444.

[7] S. Watanabe, A. Sako, and A. Nakamura, "Automatic determination of acoustic model topology using variational bayesian estimation and clustering for large vocabulary continuous speech recognition," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, pp. 855–871, 2006.

[8] T. T. Kristjansson, *Speech Recognition in Adverse Environments: a Probabilistic Approach*, Ph.D. thesis, University of Waterloo, 2002.

[9] J.T. Chien, "A bayesian prediction approach to robust speech recognition and online environmental learning," *Speech Communication*, vol. 37, pp. 321–334, 2002.

[10] H.G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR2000*, 2000.