INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Uncertainty decoding with adaptive sampling for noise robust DNN-based acoustic modeling

Dung T. Tran, Marc Delcroix, Atsunori Ogawa, Tomohiro Nakatani

NTT Communication Science Laboratories, NTT corporation,
2-4, Hikaridai, Seika-cho (Keihana Science City), Soraku-gun, Kyoto 619-0237 Japan
{dung.tran,marc.delcroix,ogawa.atsunori,nakatani.tomohiro}@lab.ntt.co.jp

Abstract

Although deep neural network (DNN) based acoustic models have obtained remarkable results, the automatic speech recognition (ASR) performance still remains low in noise and reverberant conditions. To address this issue, a speech enhancement front-end is often used before recognition to reduce noise. However, the front-end cannot fully suppress noise and often introduces artifacts that are limiting the ASR performance improvement. Uncertainty decoding has been proposed to better interconnect the speech enhancement front-end and ASR back-end and mitigate the mismatch caused by residual noise and artifacts. By considering features as distributions instead of point estimates, the uncertainty decoding approach modifies the conventional decoding rules to account for the uncertainty emanating from the speech enhancement. Although the concept of uncertainty decoding has been investigated for DNN acoustic models recently, finding efficient ways to incorporate distribution of the enhanced features within a DNN acoustic model still requires further investigations. In this paper, we propose to parameterize the distribution of the enhanced feature and estimate the parameters by backpropagation using an unsupervised adaptation scheme. We demonstrate the effectiveness of the proposed approach on real audio data of the CHiME3 dataset.

Index Terms: speech recognition, deep neural network, uncertainty decoding, adaptation

1. Introduction

Progress in acoustic modeling with deep neural network (DNNs) [1] has significantly improved the performance of automatic speech recognition (ASR). However, DNN-based acoustic models still perform poorly in adverse environments such as in the presence of noise or reverberation. The effect of noise and reverberation can be greatly mitigated by using a speech enhancement (SE) front-end prior to ASR [2, 3, 4, 5, 6]. However, this may not be sufficient as the SE front-end cannot completely remove noise or reverberation from the input signal and also often introduces distortions.

In addition to the SE front-end, uncertainty decoding at the back-end is one approach that can mitigate the mismatch caused by the residual noise and distortions at the output of the SE front-end. By considering input features as distributions, uncertainty decoding accounts for the uncertainty that emanates from the SE front-end. Uncertainty decoding for a Gaussian Mixture Model based acoustic model has been investigated intensively [7, 8, 9, 10, 11, 12, 13]. In such studies, the posterior distribution of the enhanced feature is assumed to be a Gaussian distribution where the mean is the enhanced feature and its variance represents distortion or uncertainty. An uncertainty decoding approach increases the variance of all the Gaussian components of the acoustic modeling by adding the feature uncertainty to them. The uncertainty is often hard to obtain due to the nonlinearity of the feature extraction process and the weakness of the Gaussian assumption. Non-Gaussian distributions have been shown [14, 15, 16] to be more effective than Gaussian distributions. In addition, re-estimation of the variance of the Gaussian distribution using a Maximum Likelihood estimator can greatly improve the uncertainty estimation [9] revealing the importance of the uncertainty estimator.

More recently, there have been several investigations into employing uncertainty decoding for DNNs [17, 18, 19, 20, 21, 22]. Unlike with the GMM case, it is not straightforward to include feature uncertainty in the decoding process for DNNs. Therefore, several studies have proposed using a sampling based approach i.e. averaging the DNN's output where the input features are sampled from the posterior distribution of the enhanced features. The averaging based Monte Carlo (MC) method requires many samples of the high dimensional features and this may make the approach inefficient. In addition, forwarding many samples through the DNN is computationally intensive. Moreover, samples need to be obtained from the true posterior distribution of the enhanced features in order to obtain a good estimation of the DNN output. In [23], an approach designed to improve the averaging based MC method is proposed by drawing fewer samples and weighting each DNN's output by a weight that is determined by the Minimum Classification Error criterion. [21] proposed an approach for incorporating uncertainty in both training and decoding and proposed to generating samples by interpolation between the noisy and enhanced features. Although, uncertainty estimation is crucial for uncertainty decoding, there has been little work focusing on uncertainty estimation for a DNN based acoustic model. In general, the uncertainty is usually obtained independently of the recognition, which may not be optimal.

In this paper, we propose an adaptive sampling-based approach for uncertainty decoding. The posterior distribution of the enhanced feature is explicitly parameterized, and its parameters are estimated based on the adaptation data. We parameterize the distribution by using a few important samples (usually fewer than 4) and assign weights to the DNN output for each sample. Both the weight and the samples are estimated so that they maximize the log-likelihood function of the data. Our adaptation scheme requires fewer parameters and so requires only a few utterances of data. In addition, our proposed approach can be combined with linear input network (LIN) based speaker adaptation.

In the remainder of the paper, we introduce notations and revise conventional DNN training and decoding in Section 2. Section 3 discusses the proposed uncertainty feature based adaptation approach. Some previous related studies are dis-

http://dx.doi.org/10.21437/Interspeech.2017-793

cussed in Section 4. We discuss our experimental settings and results in Section 5. Finally, Section 6 concludes the paper and presents potential future research directions.

2. Conventional DNN training/decoding

In this section, we revisit conventional neural network training and decoding for the general classification problem [24]. We assume there are pairs of training data $\mathcal{D}\,\langle \mathbf{x}_n, \mathbf{t}_n \rangle$, $n = 1, ..., N$ where N is the number of samples. $\mathbf{x}_n \in \mathcal{R}^M$ is a feature that has real values (typically log-mel filter bank coefficients) and $\mathbf{t}_n$ is a one-hot K dimensional vector that presents the corresponding class $\mathcal{C}_k$ to which the feature $\mathbf{x}_n$ belongs. Here $\mathbf{t}_n$ represents the HMM state index to which the feature belongs. For the data $\mathcal{D}\,\langle \mathbf{x}_n, \mathbf{t}_n \rangle$, the likelihood function is given by

$$p(\mathbf{t}_n|\mathbf{x}_n, \theta) = \prod_{k=1}^{K} (f_k(\mathbf{x}_n, \theta))^{\mathbf{t}_{nk}} \tag{1}$$

where $f_k(.)$ is the k^{th} element of a highly nonlinear transformation which is represented by a DNN that has parameters θ. Note that the posterior probability for each class given the input feature $\mathbf{x}_n$ is exactly the output of the corresponding DNN. It is given by:

$$P(\mathcal{C}_k|\mathbf{x}_n) = f_k(\mathbf{x}_n, \theta) \tag{2}$$

The parameter of the DNN is adjusted by maximizing the log likelihood function $p(\mathbf{t}_n|\mathbf{x}_n, \theta)$ and it is written as:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log(\prod_{n=1}^{N} p(\mathbf{t}_n|\mathbf{x}_n, \theta)), \tag{3}$$

which can be shown to be equivalent to minimizing the cross-entropy between the DNN's output and the target output. At the decoding stage, only the posterior probability for each class shown in E.q (2) is computed. To make DNNs more robust to noise and reverberation, the features $\mathbf{x}_n$ are usually computed from the distorted (noisy and reverberant) speech for the training stage and clean speech for decoding stage [25, 4]. However, at the decoding stage, it is hard to obtain the clean speech feature due to the presence of noise and reverberation which makes the ASR performance poor. In the following, we describe how to deal with noisy and reverberant speech at the decoding stage. For simplicity, we omit the notation n hereafter.

3. Proposed adaptive sampling based uncertainty decoding

When dealing with noisy speech, it is common to use an SE front-end to reduce noise prior to recognition. In addition, uncertainty decoding can be used to better interconnect an SE front-end and an ASR back-end. In this section, we briefly review the ideas behind uncertainty decoding and introduce our proposed adaptive sampling approach.

3.1. Uncertainty decoding

During testing, we observe a previously unseen noisy feature $\mathbf{y}$ in the training data. The posterior distribution of each class given the noisy feature $\mathbf{y}$ can be computed by marginalizing over the hidden variable clean feature $\mathbf{x}$:

$$p(\mathcal{C}_k|\mathbf{y}) = \int p(\mathcal{C}_k, \mathbf{x}|\mathbf{y}) d\mathbf{x}, \tag{4}$$

which can be decomposed as

$$p(\mathcal{C}_k|\mathbf{y}) = \int p(\mathcal{C}_k|\mathbf{x}) p(\mathbf{x}|\mathbf{y}) d\mathbf{x}. \tag{5}$$

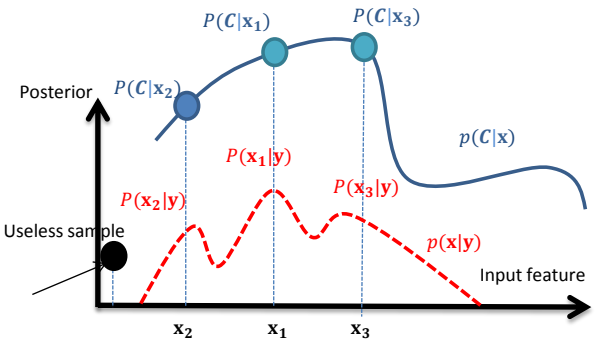

Figure 1: *Illustration of our proposed approach in case of one-dimension features: the blue curve represents the posterior distribution for each class, which is modeled by a DNN; the red dashed curve represents the posterior distribution of the enhanced features (it is unknown and must be estimated). Three one-dimension samples $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are generated and provide corresponding posterior probabilities. Note that some samples have less impact (e.g. on the most left) and therefore they could not be used to characterize the posterior distribution of enhanced features.*

Note that since $p(\mathcal{C}_k|\mathbf{x})$ is represented by a highly nonlinear function as a DNN, the above integral becomes intractable. The posterior distribution $p(\mathbf{x}|\mathbf{y})$ has a mean represented by the enhanced feature $\hat{\mathbf{x}}$ and the variance represents the uncertainty of the enhanced feature $\hat{\mathbf{x}}$. The conventional decoding simply uses the enhanced feature $\hat{\mathbf{x}}$ and it neglects the uncertainty, while the uncertainty decoding replaces the point estimate of the enhanced features $\hat{\mathbf{x}}$ with a distribution $p(\mathbf{x}|\mathbf{y})$. If the enhanced feature $\hat{\mathbf{x}}$ is well matched to the clean feature $\mathbf{x}$ (e.g good SE front-end), the posterior distribution is more peaky. Otherwise, if the enhanced feature $\hat{\mathbf{x}}$ is not well matched to the clean feature $\mathbf{x}$ (e.g. bad SE front-end), the shape of the posterior distribution becomes wider. In both cases, the type of posterior distribution is unknown. The traditional SE front-ends provide an estimation of the posterior distribution $p(\mathbf{x}|\mathbf{y})$ but this is not well matched to DNN acoustic models. In the following, we will try to approximate this posterior distribution and re-estimate it.

3.2. Adaptive sampling based uncertainty decoding

Since we do not know the behavior of the likelihood distribution, we can approximate the above posterior distribution by using a finite sum as

$$P(\mathcal{C}_k|\mathbf{y}) \approx \frac{1}{M} \sum_{m=1}^{M} P(\mathcal{C}_k|\mathbf{x}_m) P(\mathbf{x}_m|\mathbf{y}) \tag{6}$$

where M is the number of samples. By approximating the integral we can parameterize the posterior distribution $p(\mathbf{x}|\mathbf{y})$ of the enhanced feature. Then, we estimate this distribution by learning a set of discrete points $P(\mathbf{x}_m|\mathbf{y})$. Choosing good samples $\mathbf{x}_m$ is important because they can model the distribution well even with few samples. The sampling scheme is illustrated in Figure 1. In the following, for simplicity, we assume that $P(\mathbf{x}_m|\mathbf{y})$ is a scalar parameter of our model that we will esti-

mate. Each sample is generated as follows

$$\mathbf{x}_m = \hat{\mathbf{x}} + \mathbf{g}_m \odot \mathbf{d} \tag{7}$$

so that the posterior probability for a given class $\mathcal{C}_k$ is:

$$P(\mathcal{C}_k|\mathbf{x}_m) = f_k(\hat{\mathbf{x}} + \mathbf{g}_m \odot \mathbf{d}, \theta) \tag{8}$$

where $\odot$ denotes the element-wise multiplication. $\mathbf{d}$ is computed by subtracting the noisy feature from the enhanced feature e.g. $\mathbf{d} = \mathbf{y} - \hat{\mathbf{x}}$. $\mathbf{g}_m$ is a vector that is used to dynamically compensate $\mathbf{d}$ in order to obtain good samples. Note that $\mathbf{g}_m$ can includes negative value and so it allows samples to be generated that are not limited in between the noisy feature and the enhanced feature. With the proposed adaptive sampling, we estimate $\alpha_m = P(\mathbf{x}_m|\mathbf{y})$ and $\mathbf{g}_m$ using adaptation data. We first perform 1^{st} pass decoding to obtain $\mathbf{t}_n$ and then we estimate all parameters $P(\mathbf{x}_m|\mathbf{y})$ and $\mathbf{g}_m$ by maximizing the log-likelihood function (3). Since $P(\mathbf{x}_m|\mathbf{y})$ is scalar, the parameters of the posterior distribution can be estimated based on few utterances and they are used for newly arrived enhanced/distorted features. In combination with LIN based speaker adaptation, the posterior probability for each class can be given by:

$$P(\mathcal{C}_k|\mathbf{x}_m) = f_k(\mathbf{L}(\hat{\mathbf{x}} + \mathbf{g}_m \odot \mathbf{d}), \theta) \tag{9}$$

where $\mathbf{L}$ is an affine transformation. Note that, only a diagonal matrix is used for LIN based speaker adaptation.

4. Relationship to previous studies

This paper is inspired by the approach proposed in [21] where samples were manually selected. Our proposed method allows us to collect samples that maximize the log-likelihood function. The proposed approach is also related to [23]. While [23] used an MCE criterion to estimate the weights, the proposed approach uses a log-likelihood function to estimate the weights. By directly adjusting the samples, the proposed approach might avoid forwarding unreliable samples through the network. The proposed method can significantly improve performance with a small number of samples which achieves significant computation cost reduction compared with the method described in [23]. Finally, this work is also related to [22]. Whereas Nathwanni's approach requires the true uncertainty to learn the mapping to the estimated uncertainty, this work does not. Instead, the posterior distribution of the enhanced/distorted feature is learned directly on the adaptation data.

5. Experiments

5.1. Dataset

We perform experiments using the CHiME-3 corpus [26] that consists of real speech recordings collected in four different environments, i.e. cafe (CAF), street junction (STR), public transport (BUS), and pedestrian area (PED). The training corpus also includes simulated and real data sets. To evaluate our proposed approach, we discarded the simulated test data sets from our evaluation. As described in [26], speech data were recorded using a tablet device with six microphones. The corpus consists of read speech, where the prompts are taken from the WSJ0 corpus. The training set has 6 channel data in which each channel comprises 1600 real and 7138 simulated utterances, which amounts to 6*18 hours of speech. The development and evaluation sets for the real recordings consist of 1640 and 1320 utterances, respectively, spoken by four different speakers. The evaluation and development data from a given speaker cover the four different environments. In accordance with the CHiME3 challenge regulations, speaker labels can be used to perform adaptation.

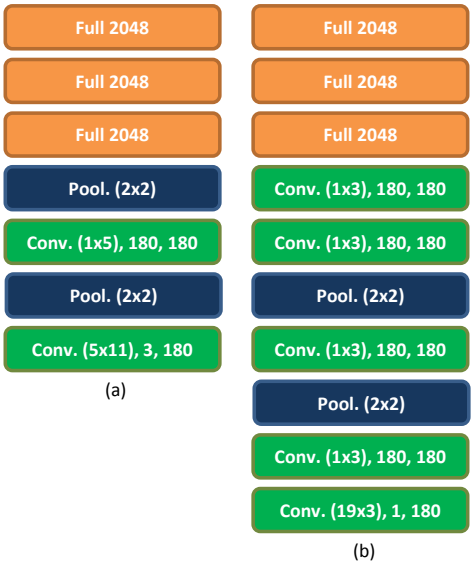

Figure 2: *Two CNN network architectures which are used for acoustic model. (a) the simple CNN; (b) the deep CNN*

5.2. Settings

5.2.1. Baseline system

Our acoustic model baseline system has two CNN architectures, namely simple CNN and deep CNN as as shown in Figure 2. A softmax layer is used to compute state posteriors. The output consists of 5976 output units corresponding to the Hidden Markov Model (HMM) states. We used sigmoid activation functions for all the hidden layers. We used different types of input features for the CNN architectures. For the simple CNN, we used speech features consisting of 40 log mel filterbank coefficients appended with static, Δ and $\Delta\Delta$ coefficients. We employed 11 concatenated speech features as the input for the CNN (1320 dimensions in total). For the deep CNN, we used static speech features consisting of 80 log mel filterbank coefficients. The context window had 19 frames of speech features. All these features were extracted with a 25-msec sliding window with a 10-msec shift. The speech features were processed with utterance level cepstral mean normalization, and further normalized using mean and variance normalization parameters calculated on the training data. Both acoustic models were trained using audio from multiple channels, i.e., multi-microphone training. Note that during training we used the noisy speech signals without any speech enhancement front-end. We trained the acoustic model using mini-batch stochastic gradient descent (SGD) to minimize the cross entropy criterion. We used an initial learning rate of 0.08, a momentum of 0.9 and a batch size of 128. We gradually reduced the learning rate when the frame accuracy did not improve for a cross validation set. The learning was stopped after 40 epochs. We used dropout regularization for all fully connected layers. For testing, we used a speech enhancement front-end to reduce noise and reverberation. Our speech enhanced front-end is described in [4, 27]. The approach consists of two steps: WPE-based dereverberation and MVDR beamforming. The acoustic beam of the MVDR is controlled using steering vectors estimated

Table 1: *WER on the development and evaluation sets for simple CNN with 40 adaptation utterances. The first four lines shows the results for the 4 speakers in the dev set. The second four lines shows the results for the 4 speakers in the eval set. The results are shown for each speaker. The best results are highlighted with bold font.*

	Baseline	LIN	ASUD	LIN+ASUD
F01	6.78	6.61	6.70	**6.55**
F04	6.33	5.69	5.96	**5.61**
M03	5.22	4.90	4.99	**4.80**
M04	6.54	**6.32**	6.39	**6.32**
Ave	6.21	5.94	6.01	**5.82**
F05	10.69	9.68	10.38	**9.50**
F06	9.08	8.41	8.74	**8.35**
M05	7.97	7.53	7.85	**6.97**
M06	9.95	9.35	9.95	**9.09**
Ave	9.42	8.74	9.22	**8.47**

Table 2: *WER for CHiME3 experiment on the development and evaluation sets for simple CNN with diferent number of adaptation utterances: 10 utterances, 20 utterances and 40 utterances. The best results is highlighted with bold font.*

Number of utterances	LIN eval (dev)	LIN+ASUD eval (dev)
10	9.18 (6.04)	8.95 (5.93)
20	8.98 (6.00)	8.75 (5.89)
40	8.74 (5.94)	**8.47** (**5.82**)

based on spectral masks. We used a trigram language model for decoding.

5.2.2. Setting for speaker adaptation and adaptive sampling based uncertainty decoding

The state alignments were computed using enhanced features and a trigram language model. The state alignments were computed separately for simple CNN and deep CNN acoustic models. For each speaker, we selected 10, 20 or 40 utterances at random from data to learn the parameter, and we use the learned parameter to decode all the data. During the adaptation, we choose different learning rates depending on the number of adaptation utterances. The learning rates were $0.0025, 0.005$ and 0.01 for 10, 20 and 40 adaptation utterances, respectively. The momentum was set at 0.9 and kept fixed for all experimental settings. The number of epochs was fixed at 10.

5.3. Results and discussion

In the remaining experiments, for which we used adaptive sampling based uncertainty decoding (ASUD) method, we found that we obtained the best results when we fixed the weight for the enhanced features to 1 and learned the two other weights form the adaptation data. We believe that this strategy also help stabilize the learning. In addition, in our implementation, we performed the weighting before the softmax layer. We keep this setup the same for both sets of experiments with simple CNN and deep CNN.

Table 3: *WER for CHiME3 experiment on the development and evaluation sets for deep CNN. The best results is highlighted with bold font.*

Number of utterances	LIN eval (dev)	LIN+ASUD eval (dev)
10	8.37 (5.37)	8.26 (5.22)
20	8.24 (5.36)	8.11 (5.15)
40	8.17 (5.32)	**7.96** (**5.12**)

5.3.1. For simple CNN

We first conduct an experiment on the simple CNN to observe the interaction between ASUD and LIN based speaker adaptations. The results in Table 1 reveal that the gains of ASUD and LIN based speaker adaptation are complementary. With only 40 utterances, the use of the ASUD approach on top of LIN based speaker adaptation obtained a 10% relative WER reduction. With only 40 utterances, LIN based speaker adaptation also provide a 7% relatively WER reduction. Although our proposed approach obtained only a small improvement compared with LIN based speaker adaptation but they are consistent. The results for both LIN based speaker adaptation and the ASUD approach with different numbers of adaptation utterances are presented in Table 2. The results show that the combination method can obtained a 5% relative WER reduction compared with the baseline with only 10 utterance (less than 1.2 minutes on the average).

5.3.2. For deep CNN

We also conducted an experiment with the deep CNN baseline. The results are shown in Table 3. With only 40 utterances, LIN based speaker adaptation also provide a 4% relative WER reduction and the ASUD approach gave a 2% relatively WER reduction to the baseline. In both experiments, we observed the values of the other two weights $P(\mathbf{x}_m|\mathbf{y})$ were much smaller than 1 revealing that the posterior distribution of the enhanced feature is peaky. We also tested the MC based approach with only three samples (the enhanced feature and two samples generated from the uniform distribution $\mathcal{U}(\hat{\mathbf{x}}, \mathbf{y})$) then we assigned the weight $1/3$ to DNN's outputs (before the softmax layer). The average of DNN output was then input into the softmax layer providing the final posterior probability for each class. We found that the MC based approach was slightly worse than the baseline. This might be because there were too few samples. The result is also consistent with the observation reported in [23].

6. Conclusions

In this paper, we investigated an approach designed to improve the interconnection of an SE front-end and a DNN acoustic model. By introducing a parameterized model of the posterior distribution of the enhanced speech feature and estimating the parameters adaptively, our proposed approach was able to find a better representation of enhanced features that matched the DNN acoustic model. The method can also be combined with speaker adaptation with a few utterances and thus allow network to adapt quickly to both speaker and noise environments. We tested our approach on the real data set of the CHiME3 challenge task and obtained promising results. In the future, we will explore the used of this approach for re-training based adaptation. In addition, studies of this adaption with other SE front-ends such as time-frequency masking are also promising. Finally, our future work will also include extensions to other training criteria such as SMBR.

7. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of fours research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?" in *INTERSPEECH*, 2013, pp. 2992–2996.

[3] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *ICASSP*, 2013, pp. 7398–7402.

[4] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *ASRU*, 2015, pp. 436–443.

[5] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, K. Kinoshita, M. Espi, S. Araki, T. Hori, and T. Nakatani, "Strategies for distant speech recognitionin reverberant environments," *EURASIP Journal on Advances in Signal Processing*, vol. 1, pp. 1–15, 2015.

[6] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, 2016.

[7] J. A. Arrowood and M. A. Clements, "Using observation uncertainty in HMM decoding," in *Proc. Interspeech*, 2002, p. 15611564.

[8] D. Kolossa and R. Haeb-Umbach, *Robust speech recognition of uncertain or missing data*. New York: Springer, 2011.

[9] M. Delcroix, T. Nakatani, and S. Watanabe, "Static and dynamic variance compensation for recognition of reverberant speech with dereverberation preprocessing," *IEEE Transatcion on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 324–334, 2009.

[10] H. Liao and M. J. F. Gales, "Joint uncertainty decoding for noise robust speech recognition," in *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*. ISCA, 2005, pp. 3129–3132.

[11] J. Droppo, L. Deng, and A. Acero, "Uncertainty decoding with splice for noise robust speech recognition," in *Proc. ICASSP*, May 2002.

[12] L. Deng, J. Droppo, and A. Acero, "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion." *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 3, pp. 412–421, 2005.

[13] A. Ozerov, M. Lagrange, and E. Vincent, "Uncertainty-based learning of acoustic models from noisy data." *Computer Speech Language*, vol. 27, no. 3, p. 874894, 2013.

[14] R. F. Astudillo, "An extension of STFT uncertainty propagation for GMM-based super-Gaussian a priori models," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1163–1166, 2013.

[15] R. C. V. Dalen and M. J. F. Gales, "Importance sampling to compute likelihoods of noise-corrupted speech," *Comput. Speech Lang.*, vol. 27, no. 1, pp. 322–349, Jan. 2013.

[16] D. T. Tran, E. Vincent, and D. Jouvet, "Nonparametric uncertainty estimation and propagation for noise robust ASR," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 23, no. 11, pp. 1835–1846, 2015.

[17] R. F. Astudillo and J. P. da Silva Neto, "Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition," in *Proc. Interspeech*, 2011, p. 461464.

[18] R. F. Astudillo, A. Abad, , and I. Trancoso, "Accounting for the residual uncertainty of multi-layer perceptron based features," in *Proc. ICASSP*, 2014, p. 68596863.

[19] A. H. Abdelaziz, S. Watanabe, J. R. Hershey, E. Vincent, and D. Kolossa, "Uncertainty propagation through deep neural networks." in *INTERSPEECH*, 2015, pp. 3556–3560.

[20] C. Huemmer, R. Maas, A. Schwarz, R. Astudillo, and W. Kellermann, "Uncertainty decoding for DNN-HMM hybrid systems based on numerical sampling." in *INTERSPEECH*, 2015, pp. 3556–3560.

[21] Y. Tachioka and S. Watanabe, "Uncertainty training and decoding methods of deep neural networks based on stochastic representation of enhanced feature," in *INTERSPEECH16*, 2015, pp. 3541–3545.

[22] K. Nathwani, J. A. Morales-Cordovilla, S. Sivasankaran, I. Illina, and E. Vincent, "An extended experimental investigation of DNN uncertainty propagation for noise robust ASR," in *5th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA 2017)*, 2017.

[23] C. Huemmer, R. F. Astudillo, and W. Kellermann, "An improved uncertainty decoding scheme with weighted samples for multichannel dnn-hmm hybrid systems," in *Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, San Francisco, 2017, pp. 31–35.

[24] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[25] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 285–290.

[26] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 504–511.

[27] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline asr in noise," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5210–5214.