HAL
open science

Discriminative importance weighting of augmented training data for acoustic model training

Sunit Sivasankaran, Emmanuel Vincent, Irina Illina

To cite this version:

Sunit Sivasankaran, Emmanuel Vincent, Irina Illina. Discriminative importance weighting of augmented training data for acoustic model training. 42th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2017), Mar 2017, New Orleans, United States. hal-01415759v2

HAL Id: hal-01415759

https://inria.hal.science/hal-01415759v2

Submitted on 6 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DISCRIMINATIVE IMPORTANCE WEIGHTING OF AUGMENTED TRAINING DATA FOR ACOUSTIC MODEL TRAINING

Sunit Sivasankaran[1,2,3], *Emmanuel Vincent*[1,2,3], *Irina Illina*[1,2,3]

[1] Inria, Villers-lès-Nancy, F-54600, France
[2] Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France
[3] CNRS, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France

ABSTRACT

DNN based acoustic models require a large amount of training data. Parametric data augmentation techniques such as adding noise, reverberation, or changing the speech rate, are often employed to boost the dataset size and the ASR performance. The choice of augmentation techniques and the associated parameters has been handled heuristically so far. In this work we propose an algorithm to automatically weight data perturbed using a variety of augmentation techniques and/or parameters. The weights are learned in a discriminative fashion so as to minimize the frame error rate using the standard gradient descent algorithm in an iterative manner. Experiments were performed using the CHiME-3 dataset. Data augmentation was done by adding noise at different SNRs. A relative WER improvement of 15% was obtained with the proposed data weighting algorithm compared to the unweighted augmented dataset. Interestingly, the resulting distribution of SNRs in the weighted training set differs significantly from that of the test set.

***Index Terms*—** ASR, data augmentation, feature simulation, DNN, CHiME.

1. INTRODUCTION

State of the art automatic speech recognition (ASR) systems are built using deep neural network (DNN) based acoustic models [1], which require large amounts of labeled training data. Obtaining this data is time-consuming and expensive. Furthermore, ensuring good performance across all acoustic conditions requires data capture across varied acoustic conditions which is practically unfeasible.

To address these issues, parametric data augmentation techniques are often employed to increase the amount and variety of data without incurring any extra labeling cost. General techniques include perturbing the vocal tract length [2], converting one speaker's data to another speaker's [2], and varying the speed or intensity [3, 4]. These techniques have shown benefit for both under-resourced languages [5] and large-scale ASR tasks [6]. A complementary way of widening the range of acoustic conditions is to simulate additional data by convolving clean speech with real or simulated acoustic impulse responses with various reverberation times (RTs) and direct-to-reverberant ratios (DRRs) [7] and adding separately recorded background noise at various signal-to-noise ratios (SNRs) [6]. This greatly increases robustness in distant-microphone conditions, as shown by the REVERB [8], ASPIRE [9], and CHiME [10, 11] challenges. Data augmentation has also been employed in other areas such as speech enhancement [12] and sound event detection [13, 14].

Not all augmented data is useful. For example, certain augmentation techniques were found to degrade the classification accuracy in [13]. The augmentation techniques and the corresponding parameters (e.g., vocal tract length factor, speed factor, SNR, RT, DRR) must be carefully chosen in order to improve performance. So far, this choice has been handled by trial and error. Considering this problem in a rigorous optimization framework, we raise the following fundamental question: "What is the optimal training set given the task, the classifier, and the test conditions?"

The literature on transfer learning provides a set of methods to answer this question, based on the assumption that the distribution of the training and test data must match [15–20]. The method in [16] operates by weighting each training sample x by the importance weight $\frac{p_{\text{test}}(x)}{p_{\text{train}}(x)}$, defined as the ratio of densities in the test and training domains. Several techniques have been proposed to estimate the density ratio [21,22]. Theoretical guarantees have been established only when the first moment of the ratio is bounded [23], which is arguably a strong restriction. More critically, there is increasing evidence that mismatched training data can outperform matched data. The first formal proof of this somewhat surprising result was published in [24]. In the field of robust ASR, it was found that, when testing on single-channel enhanced data, training on multichannel enhanced data [25], multichannel noisy data [26, 27], or both [4, 28] improves the WER despite the increased mismatch. Simulating additional data with a lower SNR than the test data can also be beneficial [29]. Even more surprisingly, not including the test condition in the training set sometimes improves the WER [11].

These results show that there is a need for a discriminative transfer learning method that maximizes performance for

the considered task, classifier, and test conditions. Such a method was proposed in [30] for kernel logistic regression. In this paper, we propose a discriminative importance weighting algorithm that is applicable to DNNs and use it to weight augmented training data for robust acoustic modeling. To do so, we assume the availability of a labeled development set with similar distribution as the test domain. We validate our approach on the CHiME-3 dataset.

The rest of the paper is organized as follows. Section 2 defines the problem and proposes an algorithm to automatically learn the importance weights. Section 3 explains the experimental setup and the results are discussed in Section 4. Section 5 concludes the paper.

2. LEARNING DATA WEIGHTS

We assume the availability of two datasets: a large augmented *training* set which has been generated by applying all possibly relevant data augmentation techniques and/or parameters, and a smaller *development* set which is distributed similarly to the (unknown) test data. We denote by $p_{\text{train}}(x)$ and $p_{\text{dev}}(x)$ the distribution of the feature vectors x in the training and development sets, respectively. Note that the two datasets are fixed: no data are added or removed by the proposed algorithm.

Classically, a DNN acoustic model with weight and bias parameters θ is trained to estimate the posterior $p_\theta(y|x)$ over the labels (senones) y given the input features x. The parameters θ are updated by stochastic gradient descent (SGD) so as to minimize the loss $\mathcal{L}$ in the training set:

$$\hat{\theta} = \arg\min_{\theta} E_{p_{\text{train}}(x)p(y|x)}[\mathcal{L}(p_\theta(y|x), y)]. \tag{1}$$

Using the cross-entropy as the loss function, the estimated loss for a batch of size N is defined as

$$E\big[\mathcal{L}(p_\theta(y|x), y)\big] = -\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(y_i|x_i). \tag{2}$$

2.1. Formulation as an optimization problem

Instead of using the training set as such, we propose to weight each training sample i by an importance weight $\omega_i \geq 0$. We incorporate these data weights as part of the loss function as

$$E\big[\mathcal{L}_\omega(p_\theta(y|x), y, \omega)\big] = -\frac{\sum_{i=1}^{N} \omega_i \log p_\theta(y_i|x_i)}{\sum_{i=1}^{N} \omega_i}. \tag{3}$$

For a given set of data weights, we can train a corresponding DNN acoustic model that minimizes the weighted loss:

$$\hat{\theta} = \arg\min_{\theta} E_{p_{\text{train}}(x)p(y|x)}\big[\mathcal{L}_\omega(p_\theta(y|x), y, \omega)\big] \tag{4}$$

The classification error $\mathcal{E}$ achieved by this DNN on the development set can then be computed as

$$\mathcal{E}(p_{\hat{\theta}}(y|x), y) = \begin{cases} 0 & \text{if } \arg\max_z p_{\hat{\theta}}(z|x) = y \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

The problem of optimizing the training set for the task, the classifier, and the test conditions translates into finding the data weights $\hat{\omega}$ for which the corresponding DNN acoustic model $\hat{\theta}$ yields minimum error rate on the development set:

$$\hat{\omega} = \arg\min_{\omega} E_{p_{\text{dev}}(x)p(y|x)}\big[\mathcal{E}(p_{\hat{\theta}}(y|x), y)\big]. \tag{6}$$

2.2. Algorithm

We optimize (4) and (6) iteratively as follows. Each iteration consists of two steps. In the first step, given the data weights ω, the DNN parameters θ are updated via one epoch of SGD on the full training set. The gradient of the weighted loss is computed by backpropagation and multiplied by $\omega_i / \sum_{i=1}^{N} \omega_i$ for each sample x_i. In the second step, given the DNN parameters θ, the data weights ω are updated using one step of gradient descent. The gradient of the classification error with respect to each data weight ω_i can be computed by updating the DNN via one step of SGD on a single sample x_i and computing the difference Δe_i between the classification errors on the development set before and after this update. The weights are then updated as $\omega_i = \omega_i - \lambda \Delta e_i$ with a suitable learning rate λ. The algorithm is summarized in Algorithm 1.

Input: Training set $\mathcal{S}_{train}$, development set $\mathcal{S}_{dev}$,
initial DNN θ_{init}, weight learning rate λ
Output: Trained DNN with parameters $\hat{\theta}_{\text{best}}$
$\omega_i \leftarrow 1,\ \forall i$;
$\hat{\theta}_{\text{best}} \leftarrow \theta_{\text{init}}$;
$\hat{e} \leftarrow E_{p_{\text{dev}}(x)p(y|x)}[\mathcal{E}(p_{\hat{\theta}}(y|x), y)]$;
$e \leftarrow \hat{e}$;
repeat
  for *i in* $\mathcal{S}_{train}$ **do**
    $\hat{\theta}_i \leftarrow \arg\min_\theta \mathcal{L}(p_\theta(y_i|x_i), y_i)$;
    (1 epoch starting from $\hat{\theta}_{\text{best}}$)
    $e_i \leftarrow E_{p_{\text{dev}}(x)p(y|x)}[\mathcal{E}(p_{\hat{\theta}_i}(y|x), y)]$;
  end
  while $e \geq \hat{e}$ **do**
    for *i in* $\mathcal{S}_{train}$ **do**
      $\Delta e_i \leftarrow e_i - e$;
      $\omega_i \leftarrow \omega_i - \lambda \Delta e_i$;
    end
    $\hat{\theta} \leftarrow$
    $\arg\min_\theta E_{p_{\text{train}}(x)p(y|x)}\big[\mathcal{L}_\omega(p_\theta(y|x), y, \omega)\big]$;
    (1 epoch starting from $\hat{\theta}_{\text{best}}$)
    $e \leftarrow E_{p_{\text{dev}}(x)p(y|x)}[\mathcal{E}(p_{\hat{\theta}}(y|x), y)]$;
  end
  $\hat{\theta}_{\text{best}} \leftarrow \hat{\theta}$;
  $\hat{e} \leftarrow e$;
until *error* e *doesn't decrease for several iterations*;

Algorithm 1: Algorithm to learn the data weights.

In practice, learning a separate weight for every sample is undesirable, as this would result in severe overfitting. To

avoid this, the dimensionality of the weight vector ω must be reduced in some way. In the following, we propose to learn a single, shared weight ω_i for every subset of samples generated using the same data augmentation parameters. At every iteration, $\hat{\theta}_i$ is computed and this weight is updated once for all samples in this subset by running one epoch of SGD over that subset.

3. EXPERIMENTAL SET UP

3.1. Data

We conducted experiments on the CHiME-3 dataset [31]. For simplicity, only the simulated training set containing 7138 utterances by 83 speakers was used. The real training set was not included[1]. Simulation was carried out by using clean Wall Street Journal (WSJ0) utterances and adding real noise backgrounds recorded in four environments: Bus, Cafe, Pedestrian area, and Street. Besides the original simulated training set provided by the challenge organizers, we generated 6 simulated training sets by decreasing or increasing the SNR by -15, -10, -5, $+5$, $+10$, or $+15$ dB for each of the 7138 utterances. We refer to the complete simulated training set of $7138 \times 7 = 49966$ utterances as the *composite dataset*.

The development set contains 1640 real and 1640 simulated utterances by 4 speakers. Evaluation is performed on the real test set, which contains 1320 utterances spoken live in real environments by 4 speakers. The noises types and the SNRs are similar across all datasets. No speech enhancement is applied on either training, development, or test data.

3.2. Algorithm settings

A GMM-HMM trained on clean WSJ0 speech was used to align the training data. The resulting alignments form the target labels (senones) for DNN training. A GMM-HMM trained on enhanced speech (using the speech enhancement baseline provided by the challenge organizers) was used to align for the development data. The resulting alignments were used to compute the classification error.

Feature-space maximum likelihood linear regression (fMLLR) features [32] were used. They were obtained by computing 13-dimensional Mel frequency cepstral coefficients (MFCCs) using a 25 ms window and 10 ms shift. The MFCCs were spliced with 3 left and 3 right context frames and decorrelated by linear discriminant analysis (LDA) [33] followed by maximum likelihood linear transform (MLLT) [34]. The transformed features were speaker normalized to obtain 40-dimensional fMLLR features, which were concatenated with 5 left and 5 right context frames to form 440 dimensional vectors given as inputs to the acoustic model.

A DNN with 7 hidden layers, 2048 sigmoid units per layer, and 1981 senone outputs was used as the acoustic model. Pretraining was performed using restricted Boltzmann machines (RBM) followed by one epoch of training. The learning rate for SGD was initially set to 0.08 and adapted as the training progressed. The minibatch size was 256.

The proposed data weighting algorithm was applied to the composite dataset. We learned a single weight for all samples generated using the same SNR decrease/increase value. In other words, we learned 7 weights: one for the original training set and one for each of the 6 generated datasets. The data weight learning rate λ was set to 0.8. The Kaldi and Theano toolkits were used to perform these experiments.

4. RESULTS AND DISCUSSION

4.1. Baselines

To start off with, an experiment was conducted using the composite dataset without weighting. A decrease in the frame error rate (FER) on the development set was observed for the first three epochs. For further epochs, even though a decrease in the training cost was observed, an increase in the FER in the development set was seen, thereby indicating overfitting (blue curve in Fig. 1). The model with best performance on the development set was used to perform ASR on the test set. A word error rate (WER) of 26.59% was observed. For comparison, we trained the acoustic model on the original simulated training dataset of 7138 utterances and obtained a WER of 29.05%. This suggest that the raw data augmentation improved the relative ASR performance by 8.5%.

4.2. WER achieved by the proposed algorithm

We then applied the proposed weight learning algorithm on the composite dataset. The FER performance for this algorithm is shown using a black curve in Fig. 1. Using real data only for development, we obtained a WER of 22.68% on the test set, that is a relative WER improvement of 14.7% and 21.9% with respect to the unweighted composite dataset and the original training set, respectively.

Table 1 presents detailed results for the various environments and using either real and simulated data or real data alone as the development set. Since the test set contains only real utterances, the distribution of the test data is expected to be more similar to real-only development data. The relative WER improvement of 0.5% obtained using real-only development data as compared to simulated+real development data supports this fact. Another interesting observation is that the relative WER improvement achieved by the proposed algorithm (with respect to the unweighted composite dataset) is similar across all noise environments when using real-only development data.

[1] It was shown in [29] that not including the real training set, which is much smaller than the simulated set, has a minor impact on the WER.

Table 1: ASR WER comparison using acoustic model trained on different train and development datasets with and without applying the weighting algorithm. Composite dataset refers to the data with {-15,-10, -5, 0, +5, +10, +15} dB. Testing was done using the real part of the CHiME-3 test dataset.

Dataset	Weighting	Type of dev dataset	# Train utterances	WER (%)				
				BUS	CAF	PED	STR	Avg.
Original	No	Simu + Real	7138	50.08	27.27	20.37	18.51	29.05
Composite	No	Simu + Real	49966	36.85	26.84	20.80	15.22	24.92
Composite	Yes	Simu + Real	49966	34.23	23.53	19.64	13.82	**22.80**
Composite	No	Real	49966	37.37	30.11	23.22	15.65	26.59
Composite	Yes	Real	49966	32.60	24.15	19.86	14.10	**22.68**

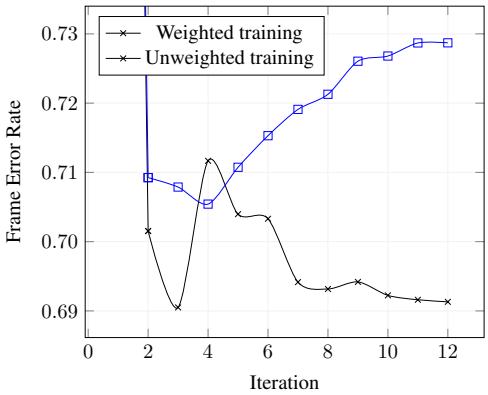

Fig. 1: FER on the development set achieved by weighted vs. unweighted training using simulated+real development data. The FER equals 0.999 for both systems at iteration 1.

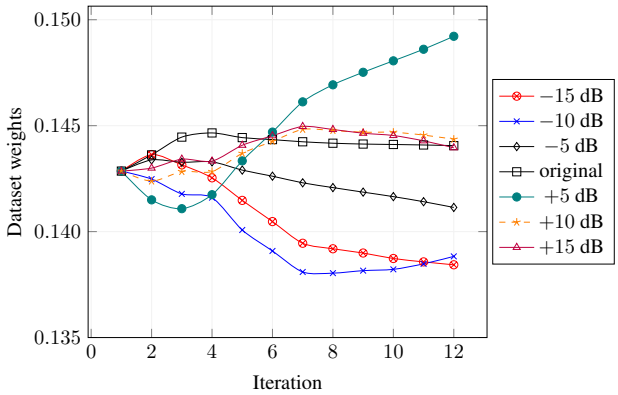

Fig. 2: Data weights learned using simulated+real development data. The sum of the weights is normalized to 1.

4.3. Learned data weights

Fig. 2 shows the evolution of the data weights per iteration. The weights for the -10, -5 and -15 dB datasets drop significantly below their initial value, while the weight for the +5 dB dataset increases significantly above its initial value. After 12 iterations, the distribution of the weighted training data is very different from that of the original training data (and from the development and test data) and it focuses on higher SNRs. This further supports the claim that mismatched training data can outperform matched data and that algorithms seeking to select matched training data [15–20] can only achieve limited success. Our algorithm was able to find a "suitably mismatched" distribution of SNRs, a result that would arguably have been hard or impossible to achieve by simple trial and error. Yet, its computational cost is only twice that of training on the unweighted composite dataset.

5. CONCLUSION

In this work we proposed an algorithm to optimize the training set for a DNN acoustic model given a development set that is representative of the test conditions. Our algorithm learns importance weights for disjoint subsets of data in the training set. The learned weights represent the importance of the data samples in each of the subsets towards minimizing the FER in the development set. Experiments were performed on the CHiME-3 dataset by simulating noisy speech with various SNRs. The results show a WER improvement of 15% relative compared to training from the unweighted dataset.

There are two future directions which we believe require further investigation. The first one is to perform a larger-scale experimental evaluation using not only noisy simulated data, but a combination of clean, enhanced, and noisy data and real and simulated data. The second one to investigate if learning weights for random subsets of data (instead of SNR specific subsets in this work) yields any realistic improvements in ASR performance. This is particularly helpful when large amount of data are available for training with no predefined demarcation information (such as SNR).

6. ACKNOWLEDGMENTS

We acknowledge the support of Bpifrance (FUI voiceHome). Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several universities as well as other organizations (see `https://www.grid5000.fr`).

7. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, A.-R. Mohamed, N. Jaitly, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 9, pp. 1469–1477, Sept. 2015.

[3] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 2440–2444.

[4] T. Schrank, L. Pfeifenberger, M. Zöhrer, J. Stahl, P. Mowlaee, and F. Pernkopf, "Deep beamforming and data augmentation for robust speech recognition: Results of the 4th CHiME challenge," in *Proc. CHiME*, 2016, pp. 18–20.

[5] A. Ragni, K. M. Knill, S. P. Rath, and M. J. F. Gales, "Data augmentation for low resource languages," in *Proc. Interspeech*, 2014, pp. 810–814.

[6] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, et al., "Deep Speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. ICML*, 2016.

[7] A. Brutti and M. Matassoni, "On the relationship between early-to-late ratio of room impulse responses and ASR performance in reverberant environments," *Speech Communication*, vol. 76, pp. 170–185, 2016.

[8] K. Kinoshita, M. Delcroix, S. Gannot, E. A. P. Habets, R. Haeb-Umbach, et al., "A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, pp. 7, Jan. 2016.

[9] M. Harper, "The automatic speech recogition in reverberant environments (ASpIRE) challenge," in *Proc. ASRU*, 2015, pp. 547–554.

[10] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Analysis and outcomes," *Computer Speech and Language*, to appear.

[11] 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*, to appear.

[12] J. Chen, Y. Wang, and D. Wang, "Noise perturbation improves supervised speech separation," in *Proc. LVA/ICA*, 2015, pp. 83–90.

[13] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," 2016, arXiv preprint arXiv:1608.04363.

[14] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," 2016, arXiv preprint arXiv:1604.07160.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[16] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[17] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. ICML*, 2004, pp. 114–121.

[18] H. Daumé III and D. Marcu, "Domain adaptation for statistical classifiers," *Journal of Artificial Intelligence Research*, vol. 26, pp. 101–126, 2006.

[19] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[20] K. Žmolíková, M. Karafiát, K. Veselý, M. Delcroix, S. Watanabe, L. Burget, and J. H. Černocký, "Data selection by sequence summarizing neural network in mismatch condition training," in *Proc. Interspeech*, 2016, pp. 2354–2358.

[21] J. Blitzer, S. Kakade, and D. P. Foster, "Domain adaptation with coupled subspaces," in *Proc. AISTATS*, 2011, pp. 173–181.

[22] K. F. Cheng and C.-K. Chu, "Semiparametric density estimation under a two-sample density ratio model," *Bernoulli*, vol. 10, no. 4, pp. 583–604, 2004.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 2012.

[24] C. R. González and Y. S. Abu-Mostafa, "Mismatched training and test distributions can outperform matched ones," *Neural Computation*, vol. 27, no. 2, pp. 365–387, Dec. 2015.

[25] Y. Fujita, T. Homma, and M. Togami, "Unsupervised network adaptation and phonetically-oriented system combination for the CHiME-4 challenge," in *Proc. CHiME*, 2016, pp. 49–51.

[26] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, et al., "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. ASRU*, 2015, pp. 436–443.

[27] S. Zhao, X. Xiao, Z. Zhang, T. N. T. Nguyen, X. Zhong, et al., "Robust speech recognition using beamforming with adaptive microphone gains and multichannel noise reduction," in *Proc. ASRU*, 2015, pp. 460–467.

[28] J. Du, Y.-H. Tu, L. Sun, F. Ma, H.-K. Wang, et al., "The USTC-iFlytek system for CHiME-4 challenge," in *Proc. CHiME*, 2016, pp. 36–38.

[29] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge," in *Proc. ASRU*, 2015, pp. 444–451.

[30] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *Proc. ICML*, 2007, pp. 81–88.

[31] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. ASRU*, 2015, pp. 504–511.

[32] M. J. F. Gales, "Maximum likelihood linear transformations for HMM based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[33] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, 2012.

[34] R. A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. ICASSP*, 1998, vol. 2, pp. 661–664.