T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| 題目(和文) | 残響および騒音のある環境下で頑健な音声認識 |
| Title(English) | Robust Speech Recognition under Reverberant and Noisy Environments |
| 著者(和文) | 太刀岡勇気 |
| Author(English) | Yuki Tachioka |
| 出典(和文) | 学位:博士(工学),<br>学位授与機関:東京工業大学,<br>報告番号:甲第10885号,<br>授与年月日:2018年3月26日,<br>学位の種別:課程博士,<br>審査員:小林 隆夫,奥村 学,山口 雅浩,金子 寛彦,篠﨑 隆宏 |
| Citation(English) | Degree:Doctor (Engineering),<br>Conferring organization: Tokyo Institute of Technology,<br>Report number:甲第10885号,<br>Conferred date:2018/3/26,<br>Degree Type:Course doctor,<br>Examiner:,,,, |
| 学位種別(和文) | 博士論文 |
| Type(English) | Doctoral Thesis |

Powered by T2R2 (Science Tokyo Research Repository)

Doctoral Dissertation

# Robust Speech Recognition under Reverberant and Noisy Environments

# 残響および騒音のある環境下で頑健な音声認識

Yuuki Tachioka
太刀岡 勇気

Feb, 2018

Department of Information and Communications Engineering,
School of Engineering, Tokyo Institute of Technology
東京工業大学 工学院 情報通信系 情報通信コース

# Contents


**1 Introduction** **1**

1.1 Background . . . 1
1.2 Remaining tasks . . . 3
1.3 Objectives of the study . . . 3
1.4 Structure of the thesis . . . 4
1.5 Acronyms in this thesis . . . 6

**2 Sound source localization methods** **9**

2.1 Introduction . . . 9
2.2 Conventional source localization methods . . . 10
2.2.1 Plane wave assumption . . . 10
2.2.2 Spherical wave assumption . . . 11
2.2.3 Cross-power spectrum phase (CSP) analysis . . . 11
2.2.4 2D-CSP method . . . 12
2.2.5 Multichannel CSP (M-CSP) method . . . 13
2.3 Prior distributions of CSP analysis . . . 14
2.3.1 Prior distributions of CSP coefficients . . . 14
2.3.2 Combination with voice activity detection information . . . 16
2.3.3 Experimental setups . . . 17
2.3.4 Results and discussion . . . 17
2.4 Template-based method for compensation of time difference of arrival (TDOA) . . . 21
2.4.1 Generalized cost function of source localization . . . 21
2.4.2 Template that modifies reference TDOA . . . 21
2.4.3 Experimental setups . . . 22
2.4.4 Results and discussion . . . 23
2.5 Conclusion of the chapter . . . 27

**3 Front-end techniques for robust automatic speech recognition (ASR)** **29**

3.1 Introduction . . . 29
3.2 Single-channel spectral-subtraction-based dereverberation . . . 30
3.2.1 Relationship between subtraction coefficients and reverberation time . . . 31
3.2.2 Estimation of reverberation time . . . 32
3.2.3 Experimental setups . . . 34
3.2.4 Results and discussion . . . 35

3.2.5 Conclusion . . . . . 36
3.3 Combination of binary masking and independent vector analysis (IVA) . . . . . 38
3.3.1 Binary masking on time-frequency domain . . . . . 38
3.3.2 Independent vector analysis using auxiliary function . . . . . 38
3.3.3 Combination of binary masking and IVA . . . . . 39
3.3.4 Experimental setups . . . . . 39
3.3.5 Results and discussion . . . . . 41
3.3.6 Conclusion . . . . . 41
3.4 Coupled initialization of spatial and spectral information for MNMF . . . . . 42
3.4.1 Matrix factorization in MNMF . . . . . 43
3.4.2 Multiplicative update rule . . . . . 43
3.4.3 ASR-based initialization of speech bases . . . . . 44
3.4.4 Initialization of spatial correlation matrices . . . . . 45
3.4.5 Coupled initialization via cluster-indicator latent variables . . . . . 46
3.4.6 Experimental setups . . . . . 46
3.4.7 Results and discussion . . . . . 47
3.4.8 Conclusion . . . . . 49
3.5 Voice activity detection (VAD) method based on likelihood ratio test . . . . . 50
3.5.1 Likelihood ratio test (LRT) . . . . . 51
3.5.2 Density ratio estimation (KLIEP) . . . . . 52
3.5.3 Application of KLIEP for VAD . . . . . 53
3.5.4 Combination of VAD systems . . . . . 54
3.5.5 Automatic thresholds determination . . . . . 54
3.5.6 Experimental setups . . . . . 55
3.5.7 Results and discussion . . . . . 56
3.5.8 Conclusion . . . . . 58
3.6 ASR performance estimation of clipped speech . . . . . 59
3.6.1 Clipped signals and clipping level . . . . . 59
3.6.2 Signal-to-noise ratio (SNR) estimation of clipped speech . . . . . 59
3.6.3 ASR performance estimation by logistic regression . . . . . 60
3.6.4 Experimental setups . . . . . 61
3.6.5 Results and discussion . . . . . 61
3.6.6 Conclusion . . . . . 63
3.7 Compensation of mismatched sampling frequency . . . . . 64
3.7.1 Gaussian-mixture-model-based bandwidth extension (BWE) . . . . . 64
3.7.2 Long short-term memory recurrent-neural-network(RNN)-based BWE . . 64
3.7.3 Experimental setups . . . . . 65
3.7.4 Results and discussion . . . . . 66
3.7.5 Conclusion . . . . . 68
3.8 Conclusion of the chapter . . . . . 68

**4 Back-end techniques for robust ASR 69**
4.1 Introduction . . . 69
4.2 ASR system overview . . . 70
4.2.1 GMM-HMM ASR systems . . . 70
4.2.2 DNN-HMM hybrid ASR systems . . . 73
4.2.3 Feature adaptations . . . 74
4.3 Linear discriminant analysis (LDA) of acoustic features . . . 76
4.3.1 Conventional maximum likelihood LDA . . . 77
4.3.2 Sequential maximum mutual information LDA . . . 78
4.3.3 Experimental setups . . . 80
4.3.4 Results and discussion . . . 80
4.3.5 Conclusion . . . 83
4.4 Discriminative training methods of acoustic models . . . 84
4.4.1 Cross-entropy (CE) training of DNNs . . . 84
4.4.2 MMI discriminative training . . . 84
4.4.3 MMI discriminative training of GMMs . . . 85
4.4.4 MMI discriminative training of DNNs . . . 85
4.4.5 sMBR discriminative training of DNNs . . . 86
4.4.6 Feature-space MMI discriminative training . . . 87
4.5 Discriminative training methods of low-rank DNN . . . 89
4.5.1 Reducing DNN model size singular value decomposition (SVD) . . . 89
4.5.2 Combination of discriminative training with SVD . . . 90
4.5.3 Experimental setups . . . 91
4.5.4 Results and discussion . . . 92
4.5.5 Conclusion . . . 93
4.6 Discriminative training methods of system combination . . . 95
4.6.1 Generalized discriminative training framework for complementary systems 96
4.6.2 Complementary acoustic model training . . . 96
4.6.3 Complementary discriminative feature transformation . . . 99
4.6.4 Experimental setups . . . 99
4.6.5 Results and discussion . . . 101
4.6.6 Conclusion . . . 103
4.7 Discriminative training methods of language models . . . 104
4.7.1 Discriminative language modeling . . . 105
4.7.2 RNN-LM and cross-entropy (CE) training . . . 105
4.7.3 Discriminative training of RNN-LM . . . 107
4.7.4 Experimental setups . . . 108
4.7.5 Results and discussion . . . 109
4.7.6 Conclusion . . . 110
4.8 Uncertainty training and decoding methods of DNN . . . 112

4.8.1 DNN uncertainty decoding . . . 113
4.8.2 DNN uncertainty training . . . 114
4.8.3 Stochastic process for the linear-interpolation coefficient . . . 115
4.8.4 Experimental setups . . . 115
4.8.5 Results and discussion . . . 116
4.8.6 Conclusion . . . 118
4.9 Conclusion of the chapter . . . 119

**5 Development of ASR systems for realistic noisy and reverberant environments 121**
5.1 Introduction . . . 121
5.2 Noisy ASR in house (The second CHiME challenge) . . . 121
5.2.1 System overview . . . 123
5.2.2 Prior-based binary masking . . . 123
5.2.3 Minimum Bayes risk decoding . . . 125
5.2.4 Combination of minimum Bayes risk decoding with discriminative language modeling . . . 125
5.2.5 System combination . . . 126
5.2.6 Experimental setups . . . 126
5.2.7 Results and discussion . . . 128
5.2.8 Conclusion . . . 133
5.3 Noisy ASR in public spaces 1 (The third CHiME challenge) . . . 135
5.3.1 System overview . . . 136
5.3.2 Speech enhancement . . . 136
5.3.3 Optimal ASR system selection based on an estimated WER via i-vector similarities . . . 137
5.3.4 Experimental setups . . . 139
5.3.5 Results and discussion . . . 140
5.3.6 Conclusion . . . 143
5.4 Noisy ASR in public spaces 2 (The fourth CHiME challenge) . . . 144
5.4.1 System overview . . . 144
5.4.2 Experimental setups . . . 145
5.4.3 Results and discussion . . . 145
5.4.4 Conclusion . . . 147
5.5 Reverberant ASR in various rooms (The REVERB challenge) . . . 148
5.5.1 System overview . . . 149
5.5.2 Speech enhancement . . . 151
5.5.3 Delay-and-sum BF with CSP analysis . . . 151
5.5.4 Speech recognition . . . 152
5.5.5 Experimental setups . . . 152

5.5.6 Results and discussion . . . 155
5.5.7 Conclusion . . . 161
5.6 Source localization and VAD in house (The DIRHA challenge) . . . 162
5.6.1 System overview . . . 162
5.6.2 Switching-Kalman-filter-based VAD . . . 163
5.6.3 Ensemble integration of calibrated speaker localization and statistical VAD 163
5.6.4 Experimental setups . . . 164
5.6.5 Results and discussion . . . 167
5.6.6 Conclusion . . . 168
5.7 Conclusion of the chapter . . . 168

**6 Conclusion 169**
6.1 Findings of each chapter . . . 169
6.2 Future work . . . 170
6.3 Thesis summary in Japanese . . . 171

**Acknowledgement 173**

**References 175**

**Publication list 199**

# 1 Introduction

## 1.1 Background

Since ancient times, speech has been an important research topic because it is the most common method of communication and many language scholars have studied speech [1, 2, 3]. Their achievements such as phonetics provide a foundation for language education. Engineering advancements in the domain of speech, particularly in automatic speech recognition (ASR), started approximately 30–40 years ago [4]. Research on ASR started with isolated word recognition [5] and was interrupted at some occasions. Rule-based methods such as spectrogram reading were used to recognize isolated words because computer resources were extremely limited, but the poor performance of these methods prevented the development of a practical ASR system.

With the rapid development of computers in 80's and 90's, new statistical methods such as dynamic programing [6, 7], Viterbi algorithm [8], hidden Markov model (HMM), neural network (NN) [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], and probabilistic language model [17] were proposed that significantly improved the ASR performance. These methods were expected to solve simple practical tasks such as digit recognition, which owes to the support of Defense Advanced Research Projects Agency (DARPA) [20]. In the 90's, academic studies were still active and applications were sought but ASR had not been widely used.

However, the increasing prevalence of car navigation systems promoted the use of ASR systems among common users. Fig. 1.1 shows the increasing trend in an annual shipment of car navigation systems since the mid 2000's because most new cars were being equipped with such systems. ASR is one of the most effective human-machine interfaces for can navigation systems because

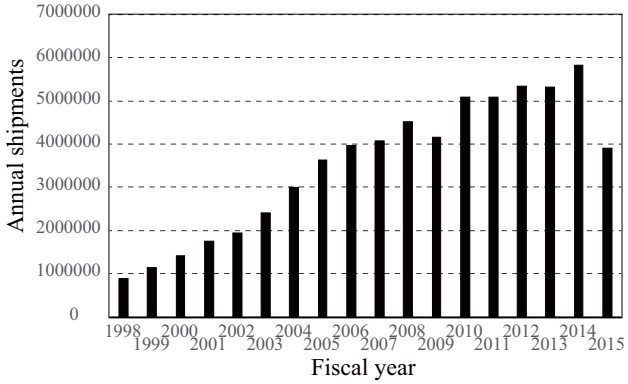



Fig. 1.1 Annual shipments of car-navigation systems in Japan.

drivers cannot resort to visual and tactile senses to control it. In such situations, speech input is an effective solution, thus leading to many companies producing ASR-based systems, although their use was limited to the specific word recognition due to the poor computational resources, which required the commands to be accurately remembered. Eventually, the number of users of ASR-based systems gradually increased with an increasing variety of acceptable commands and allowing address search for the whole country. Consequently, during this period, proficient experts and financial asserts were fostered for the development of the next generation ASR systems. However, a relatively small number of car navigation systems equipped ASR and prominent ASR applications were mainly for business-to-business (B2B) purposes, such as call center recordings and automatic generation of meeting minutes. The author developed ASR systems for elevators[1] but unfortunately these types of systems were not widely used. At this time, many people were unaware of the usefulness of ASR systems. This is because the ASR systems were mainly developed for embedded systems and the performance was lower than expected due to the system restrictions.

The advent of smart-phones completely changed this situation. An alternative input method was required, because of the lack of keyboards on smart-phones resulting in longer typing times than personal computers. The major players in web search engines and mobile phones have competed to improve ASR performance, and their promotional campaigns have widely increased the use of ASR. The development of speech enhancement technologies allowed ASR to work in highly noisy environments. Before then, single-channel inputs that could not exploit spatial information and had poor noise reduction performance for non-stationary noise were primarily used. Multi-channel inputs allowed spatial information to be used by the phase difference between microphones and thus aided in dealing with directional noises. The recent availability of multi-channel inputs in car navigation systems and smart-phones has greatly improved the noise reduction performance. Local systems require small computational loads but the connection to the internet permitted the use of complex signal processing methods and bigger models for ASR by exploiting background server resources. In particular, the introduction of deep learning in 2013 significantly enhanced the ASR performance, resulting in understanding the potency of ASR by common users. This enabled stress-free ASR applications under real environments and made the ASR technology widely known.

Deep learning is a revisit of NN-based approach, which migrates the conventional Gaussian mixture model (GMM)-based approach to the deep neural network (DNN)-based approach. At the dawn of ASR, various studies were performed to compare the performance of NNs and HMMs [12, 21]. Since 2013, NN-based systems have been re-evaluated and combined with HMMs, instead of using only NNs for recognition. Recently, the studies that introduce time structures into NNs (starting from [15, 22]) have been re-evaluated. Many researchers and engineers are struggling to find new use for ASR. Recent devices such as smart-speakers are being extensively used in home environments.

[1] http://www.mitsubishielectric.co.jp/news/2011/pdf/0303-b.pdf (Dec/04/2017 confirmed)

## 1.2 Remaining tasks

As mentioned above, ASR is now widely used, and its application will be further extended by completing two major remaining tasks. The first task is the improvement of ASR performance, especially for distant recordings and for use in highly noisy environments. Initially, ASR mainly used speech recorded by close-talking microphones in noise-less conditions. Even in noisy conditions, when close-talking microphones are used such as smart-phone cases, the high ASR performance can be achieved because of their high signal-to-noise ratio (SNR). However, the number of cases of ASR by distant microphones without close-talking microphones is increased. The emerging examples are conversation ASR systems using a system located at the center of a conference room or in-car ASR systems under highly noisy situations using microphones located on the dashboard. Model-based statistical methods used for ASR perform well for the matched conditions where training and testing environments are matched but their performance significantly deteriorates in the case of mismatched cases, e.g., when speakers with strange voices use ASR or ASR is used under mismatched noise conditions.

The second task is the realization of a natural speech interface. In recent car navigation systems, ASR can be used without necessarily remembering all the variety of acceptable commands; however, spontaneous requirements of users are still difficult for the ASR system to comprehend. Once accustomed to an ASR system, people can naturally use it; however, this is a barrier for novices who give up ASR after a few attempts. Including intention understandings in ASR systems would make the process of ASR similar to the process of asking another human and would widen its applicability. Even in the case of smart-phones, people want to have a human-like dialog with the machine, which is not necessarily goal-oriented. For these purposes, in addition to the above-mentioned intention understanding, knowledge of wide genre is required to answer user questions. Non-verbal information such as intonation can be effectively used for understanding the intension of the users. To achieve this goal, ASR can use a process different from the one used by humans [23]. The flexibility of spontaneous speech recognition can be improved by a cooperation of ASR with dialog control and language processing. Word error rate (WER) reduction is not a simple solution [24].

## 1.3 Objectives of the study

This thesis proposes some methods to improve the ASR performance under noisy and reverberant environments, which is the first remaining task. In many cases, the performance in such environments can be enhanced by preparing various types of training data, i.e., the number of speakers in training data is increased as much as possible and many types of noisy environments are prepared in order to create nearly matched conditions. Although the performance of ASR depends on human resources and computer power, the systems collecting most training data can achieve the highest accuracy. This is a simple solution but is not always sufficient.

The reason for ASR performance degradation in real environments is distortion of speech data due to various environmental factors. Distortions are classified into two types: additive and multiplicative. Noise is an additive distortions and it is an addition of speech and noise in the spectral domain. To improve ASR performance, it is necessary to eliminate noise. This thesis proposes noise reduction methods using multiple microphones to localize the speaker position and enhance speech by using phase differences between the microphones.

Multiplicative distortions are related to the path of communication and room reverberations and are a multiplication of speech and impulse responses in the spectral domain and are a convolution in the time domain. Conventional studies mainly focus on additive distortions but the performance of distant is significantly degraded by multiplicative distortions due to room reverberation. Multiplicative distortions can not necessarily be eliminated by conventional noise reduction methods. Additional speech enhancement, dereverberation, is required after the estimation of reverberation included by speech. This thesis proposes dereverberation method with reverberation time estimation.

Once noise and reverberation can be eliminated to some extent, it is necessary to detect speech activation accurately. Missing speech of course disables ASR but too much noise also degrades ASR performance; hence, an accurate voice activity detection (VAD) system is needed. This thesis also presents an effective VAD method.

Finally, the detected speech is recognized. A noise robust ASR is needed because the influence of noise persists even after its reduction. This can be achieved by discriminative training, which minimizes its objective function related to the error rates of ASR. This thesis applies discriminative training approaches to various models including feature transformation, acoustic, and language models.

The proposed methods should be validated on the realistic data. This thesis also illustrates the various challenges related to the development of a noise-robust ASR and the validation of the proposed methods in terms of WER, which clarifies the effectiveness of the proposed methods.

## 1.4 Structure of the thesis

This thesis describes the techniques[2] required to develop a noise-robust ASR system operating in the real environments. Fig. 1.2 shows the structure of this thesis. The first step is localizing a sound source. Chapter 2 introduces the conventional source localization algorithm in Section 2.2. Localization performance can be improved by using prior distributions of source direction and VAD (2.3). Template-based methods can compensate reverberation and localization errors of microphones (2.4).

In Chapter 3, two speech enhancement (SE) methods including both dereverberation and noise reduction are proposed, namely, a single-channel dereverberation method (3.2) and multi-channel method that combines noise reduction using physical information with blind source separation

[2]Sections 2.3, 2.4, 3.2, 3.3.3, 3.4.3, 3.5.3, 3.5.4, 3.5.5, 3.6.2, 3.6.3, 3.7.2, 4.3.2, 4.5.2, 4.6, 4.7.3, and 4.8 describe the newly proposed methods.

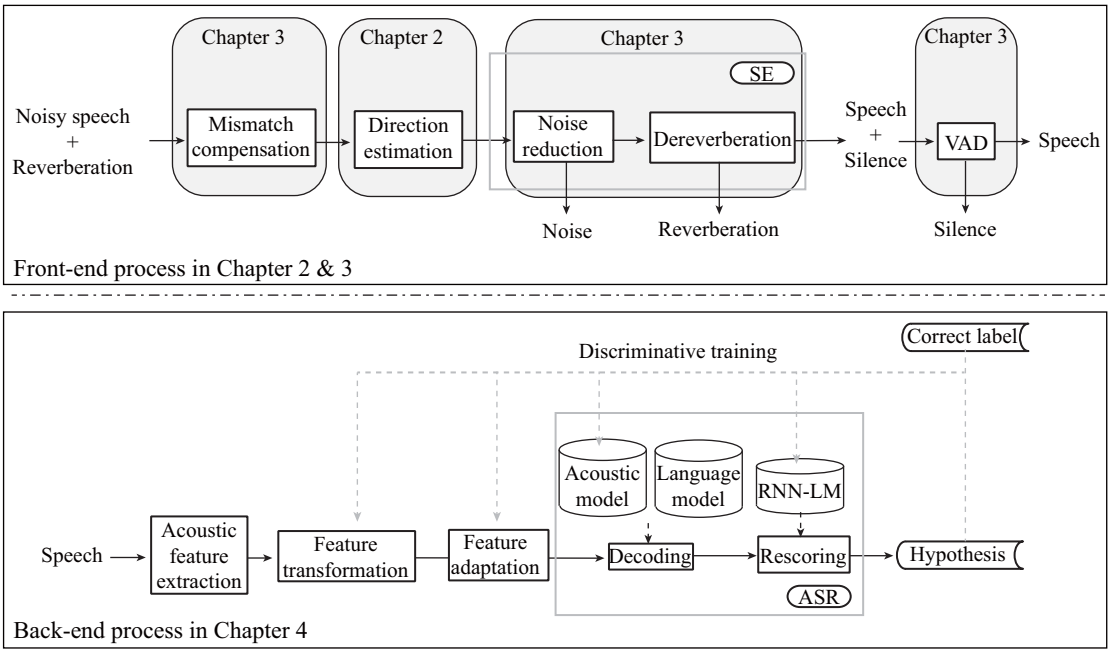



Fig. 1.2 Structure of the thesis (SE: speech enhancement, VAD: voice activity detection, RNN-LM: recurrent neural network language model).

(3.3). In addition, effective initial value setting methods for multi-channel non-negative matrix factorization (NMF) are discussed (3.4). Then, Section 3.5 describes a VAD method to detect speech activation from enhanced speech. In addition, it is necessary to address the problems due to the mismatch in recording conditions. In Section 3.6, the degradation of speech SNR and ASR performance due to clipping, which is caused by inappropriate recording levels, is evaluated. Section 3.7 evaluates the degradation of ASR performance due to sampling frequency mismatches and proposes a band-width-extension method to reduce mismatches.

Chapter 4 is focused on noise robust ASR. Section 4.2 presents an overview of ASR methods and describes the discriminative training methods in detail, which are the main focus of this chapter. Discriminative training methods use objective functions that reduce ASR errors for model training. The proposed discriminative methods are effective for feature transformation for ASR (4.3.2), acoustic models (4.4.2 and 4.5.2), and language models (4.7). This training approach is extended in Section 4.6 by proposing a framework that constructs complementary systems to be used when outputs of multiple systems are combined. Finally, the improvement of the robustness of ASR models against speech distortions due to SE is proposed in Section 4.8.

Chapter 5 validates the proposed methods on various challenges, which target noise- and reverberation-robust ASR. The CHiME series (5.2, 5.3, and 5.4) mainly targets additive noise. For this task, SE methods and noise-robust ASR systems were developed. REVERB challenge (5.5) mainly targets multiplicative noise, i.e., reverberation, and validates the effectiveness of the proposed dereverberation method. Section 5.6 describes the DIRHA challenge that evaluates the

performance of source localization and VAD. The participation to these challenges clarifies the effectiveness of the proposed method on these benchmark tasks.

## 1.5 Acronyms in this thesis

Acronyms appeared in this paper are shown in Table 1.1.

Table 1.1: Acronyms appeared in this thesis.

| | |
|---|---|
| ASR | automatic speech recognition |
| BF | beamformer |
| BM | binary masking |
| BWE | bandwidth extension |
| CMN | cepstrum mean normalization |
| CSP | cross-power spectrum phase |
| CE | cross-entropy |
| DLM | discriminative language modeling |
| DNN | deep neural network |
| DOA | direction of arrival |
| fMLLR | feature-space maximum likelihood linear regression |
| GMM | Gaussian mixture model |
| HMM | hidden Markov model |
| ICA | independent component analysis |
| IVA | independent vector analysis |
| LDA | linear discriminant analysis |
| LM | language model |
| LVCSR | large-vocabulary continuous speech recognition |
| MBR | minimum Bayes risk |
| MFCC | mel-frequency cepstrum coefficient |
| ML | maximum likelihood |
| MLLT | maximum likelihood linear transformation |
| MLLR | maximum likelihood linear regression |
| MMI | maximum mutual information |
| MPE | minimum phoneme error |
| NMF | non-negative matrix factorization |
| PLP | perceptual linear prediction |
| RNN | recurrent neural network |
| RT | reverberation time |
| SAT | speaker adaptive training |
| SE | speech enhancement |

| | |
|---|---|
| SNR | signal-to-noise ratio |
| SS | spectral subtraction |
| STFT | short-time Fourier transform |
| SVD | singular value decomposition |
| TDOA | time difference of arrival |
| VAD | voice activity detection |
| WER | word error rate |

# 2 Sound source localization methods

## 2.1 Introduction

Sound source localization and VAD are important and effective techniques for distant applications. One such application is ASR using distant microphones, e.g., in home devices. Under such conditions, it is necessary to enhance the target speech. Although there are many 'blind' speech enhancement methods solely exploiting speech characteristics [25], the additional use of speakers' positions has been shown to improve robustness and effectiveness [26, 27] over blind approaches. For example, speaker localization techniques can effectively suppress directive noise.

Sound source localization techniques expand the applicability of various applications. One of the applications is surveillance[28]. Source localization based on sound is suitable for low-cost surveillance because switching the camera to the estimated sound direction widens the covering area of surveillance. Such source localization techniques are classified into passive and active ones. Passive techniques, which only use receivers and this paper deals with, are more practical but more complicated than active ones, which use both receivers and transmitters.

To estimate the direction of arrival (DOA), the cross-power spectrum phase (CSP) analysis that uses two microphones is widely known as an effective estimation method [29, 30, 31, 32, 33]. However, the accuracy of the CSP analysis decreases at low SNR and owing to directional noise because the DOA is estimated from the peak of CSP coefficients, which is easily masked by noises. Denda *et al.* proposed a weighted CSP analysis, that weighs the spectrum in speech bands, and CSP coefficient subtraction that subtracts estimated noise components from CSP coefficients [34]. Estimation accuracy decreases when noise components are intensive in speech bands or are non-stationary. These methods are applied only for speech with stationary noise. Nishiura *et al.* proposed synchronous addition of a CSP coefficient, which uses three or more microphones and synchronously adds paired CSP coefficients [35, 36]. This increases device size and computational load. In Section 2.3, we propose a CSP analysis using prior distributions of source direction and VAD information in order to eliminate noise from CSP coefficients [37]. This method can be adopted for any sound sources if activity of the source can be detected somehow, and requires minimal computation because the algorithm is simple and realized by two microphones.

For the case that the target is limited to direction estimation, high accuracies have been achieved by above-mentioned method. In addition to the direction, if the source position can be estimated, it broadens the possibility of applications using localization[1]. Direction estimations

[1]In this paper, source localization is limited to the horizontal plane because vertical (height) estimation is

have one unknown variable under the plane wave assumption, whereas source position estimations have two or three unknown variables under the spherical wave assumption [39, 40]. The latter is much more difficult than the former. The experiments reported in Section 2.4.3 show that the tolerance errors of the latter estimation are much smaller than those of the former estimation and that without reverberation and noise, performance is high, but reverberation and noise degrade the performances of the conventional methods due to reflected sounds and measurement errors. For passive systems, to reduce their influence, some calibrations are needed[41]. Source localization methods compare an observed time difference of arrival (TDOA) with a reference TDOA. Reference TDOAs are based on the assumptions above and assume no reflected sounds and measurement errors; thus, errors degrade the source localization accuracies. To address this problem, in Section 2.4, we propose a template-based method that modifies reference TDOAs according to reference measurements [42].

## 2.2 Conventional source localization methods

The behavior of the sound propagation depends on the distance, which is relative to the wavelength and the width of microphone array, from the source. In near fields where the condition

$$\rho < \frac{2D^2}{\lambda}, \tag{2.1}$$

is satisfied, sounds propagate as a spherical wave as show in Section 2.2.2; otherwise, in far fields, sounds propagate as a plane wave as shown in Section 2.2.1 [43]. Here, $\rho$ is the distance from the center of the microphone array to the source, $D$ is the maximum width of the microphone array and $\lambda$ is the wavelength.

### 2.2.1 Plane wave assumption

In far fields, a sound source is assumed to be line that has an infinite size. The source direction $\theta_S$ is the only parameters to determine source positions. When the position of the $n$th microphone among $N$ microphones ($1 \leq n \leq N$) is $\boldsymbol{r}_n$, the TDOA $\tau^{pln}_{n_1,n_2}$ between microphones $n_1$ and $n_2$ is represented as

$$\tau^{pln}_{n_1,n_2}(\boldsymbol{s}) = \frac{|\boldsymbol{r}_{n_1} - \boldsymbol{r}_{n_2}| \sin(\theta_S)}{c}, \tag{2.2}$$

where $c$ is the sound speed.

less important. One of the applications that need height localization is a robot[38]. For height localization, microphones located at different heights are needed and this is a frequent setting for robot applications. However, a general microphone array has microphones that are located at the same height and, for these applications, height localization is less useful.

### 2.2.2 Spherical wave assumption

In near fields, a sound source is assumed to be point, and at points with the same distance from the source, the phases are identical. For example, for a frequency of 1 kHz, $\rho$ is 0.52 [m] when $D = 0.3$ [m] and $\rho$ is 2.1 [m] when $D = 0.6$ [m]. When the source position is $\boldsymbol{s}$, the TDOA $\tau^{sph}_{n_1,n_2}$ between microphones is represented as

$$\tau^{sph}_{n_1,n_2}(\boldsymbol{s}) = \frac{|\boldsymbol{r}_{n_1} - \boldsymbol{s}| - |\boldsymbol{r}_{n_2} - \boldsymbol{s}|}{c}. \tag{2.3}$$

### 2.2.3 Cross-power spectrum phase (CSP) analysis

The time-domain $s$th sample $z_n(s)$ observed by the $n$th microphone is transformed into the short-time Fourier transform (STFT) spectrum. The spectrum $X_n(t,k)$ at the $t$th frame and the $k$th frequency bin ($1 \leq k \leq K$) is obtained as

$$X_n(t,k) = \sum_{s=0}^{K-1} [\phi(s) z_n(\varphi \cdot t + s)] \exp\left(-2\pi\jmath \frac{s}{K} k\right), \tag{2.4}$$

where $\varphi$ is a frame shift, and $\phi$ is a window function with the window length $K$.

In the CSP analysis, the DOA is estimated from the arrival time delay $\tau$ [s] using a cross-power spectrum between the two microphones. First, CSP coefficients are calculated as

$$CSP_t(\tau) = \mathcal{F}^{-1}\left(\frac{\boldsymbol{X}_{n_1}(t) \odot \boldsymbol{X}_{n_2}(t)^*}{|\boldsymbol{X}_{n_1}(t)||\boldsymbol{X}_{n_2}(t)|}\right), \tag{2.5}$$

where $\boldsymbol{X}_n(t) = [X_n(t,1), ..., X_n(t,k), ..., X_n(t,K)]^\top$ is a vector form of the spectrum; $\mathcal{F}$ is a short-time Fourier transform; * denotes the complex conjugate and $\odot$ denotes the element-wise multiplication of two vectors. Here, $^\top$ denotes the transpose. CSP coefficients are a function of the delay time $\tau$ ($0 \leq \tau \leq \tau_{max} = \lfloor |\boldsymbol{r}_{n_1} - \boldsymbol{r}_{n_2}| f_s/c) + 1 \rfloor$) where $f_s$ denotes a sampling frequency [Hz] and $c$ denotes a sound speed [m/s].

Second, the arrival time delay $\tau$ of the $t$th frame is represented as a peak of CSP coefficient and is calculated as follows [29].

$$\tau^{csp}_t = \arg\max_{\tau} \left(CSP_t(\tau)\right). \tag{2.6}$$

Finally, $\theta_S$ is obtained according to Eq. (2.2) as

$$\theta_S = \sin^{-1}\left(\tau^{csp} \frac{c}{f_s |\boldsymbol{r}_{n_1} - \boldsymbol{r}_{n_2}|}\right). \tag{2.7}$$

Fig. 2.1 shows the relationship of $d_m = |\boldsymbol{r}_{n_1} - \boldsymbol{r}_{n_2}|$ and $\theta_S$.

To estimate the peak of CSP coefficients with higher accuracy, we use the second-order polynomials determined by adjacent three CSP coefficients $y_1 = CSP(\tau - 1)$, $y_2 = CSP(\tau)$, and $y_3 = CSP(\tau + 1)$ to interpolate the maximal value $\tau'$, as in Eq. (2.8):

$$\tau' = \tau - \frac{y_3 - y_1}{2(y_3 - y_2 + y_1)}. \tag{2.8}$$

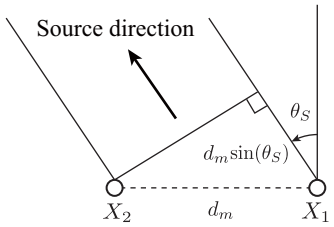



Fig. 2.1 A microphone setting.

When $\tau$ is zero or $k_{max}$, the maximal value is interpolated using the first-order polynomials on the one side. For example, when $\tau$ is zero, we interpolate $\tau'$, as in Eq. (2.9):

$$\tau' = 1 - \frac{CSP(0)}{CSP(1) + CSP(0)}. \tag{2.9}$$

### 2.2.4 2D-CSP method

The conventional source localization methods below are based on a spherical wave assumption. Although there are many source localization methods, Hayashida *et al.*[40] showed that the 2D-CSP method (in this section) and multichannel CSP method (in the next section) have high accuracies.

The original CSP method [29] only estimates the direction of arrival under the plane wave assumption. Under the condition that the microphones are distributed over a broad area, the source locations can be estimated using triangulation, but this is an unrealistic assumption. The 2D-CSP method estimates $\boldsymbol{s}$ under the spherical wave assumption[39]. To do this, the number of microphones, $N$, must be 3 or more. For example, there are two microphone pairs: $\varphi(1) = \{1, 2\}$ and $\varphi(2) = \{3, 4\}$. Here, for simplicity, the microphone intervals are the same. In the plane-wave case, $|d_1 - d_2| = |d_3 - d_4|$; thus, there is no difference in the TDOA between the microphone pairs. On the other hand, in the spherical-wave case, using the difference between $|d_1 - d_2|$ and $|d_3 - d_4|$, the distance to the sources can be estimated. The theoretical TDOAs $\tau^{sph}_{n_1,n_2}$ are determined using Eq. (2.3), whereas experimentally, the TDOAs $\tau^{csp}_{n_1,n_2}$ can be obtained by the CSP method using Eq. (2.6). Some candidate source points $\mathcal{S}$ are prepared in advance. For each candidate point $\boldsymbol{s} \in \mathcal{S}$, a cost function $P(\boldsymbol{s})$ is calculated by adding the differences between the theoretical TDOAs and the observed TDOAs of $M$ microphone pairs ($2 \leq M \leq {}_N C_2$). If the theoretical TDOAs are near to the observed TDOAs, the cost function $P$ will be small. If the measurement errors are sufficiently small, the source positions can be estimated at the minimum cost of $P(\boldsymbol{s})$ as

$$\arg\min_{\boldsymbol{s}\in\mathcal{S}} P(\boldsymbol{s}) = \arg\min_{\boldsymbol{s}\in\mathcal{S}} \sum_{m=1}^{M} \left( \tau^{sph}_{\varphi(m)}(\boldsymbol{s}) - \tau^{csp}_{\varphi(m)} \right)^2, \tag{2.10}$$

where $\varphi(m)$ is the $m$th microphone pair. Note that because one microphone pair can only indicate that a sound source exists on a hyperbola, two or more different microphone pairs (i.e., three or more microphones) are needed.

### 2.2.5 Multichannel CSP (M-CSP) method

The 2D-CSP method adds the differences in the TDOAs for each microphone pair, whereas the M-CSP method considers the differences in the TDOAs of all microphone pairs simultaneously by calculating the all-pair correlation matrix as

$$\boldsymbol{R}_k = \begin{bmatrix} \xi_{1,1,k} & \cdots & \xi_{1,N,k} \\ \vdots & \ddots & \vdots \\ \xi_{N,1,k} & \cdots & \xi_{N,N,k} \end{bmatrix}, \tag{2.11}$$

and compares this correlation matrix with given steering vectors[40]. This simultaneous consideration of the correlation between each microphone pair improves the accuracy. Each component is represented as

$$[\xi_{n_1,n_2,1}, \cdots, \xi_{n_1,n_2,K}]^\top = \frac{\boldsymbol{X}_{n_1}(t) \odot \boldsymbol{X}_{n_2}(t)^*}{|\boldsymbol{X}_{n_1}(t)||\boldsymbol{X}_{n_2}(t)|}. \tag{2.12}$$

The steering vector $\boldsymbol{a}_k$ for $\boldsymbol{s}$ is obtained as

$$\boldsymbol{a}_k(\boldsymbol{s}) = \left[e^{-\jmath\omega_k|\boldsymbol{r}_1-\boldsymbol{r}_2|/c}, \ldots, e^{-\jmath\omega_k|\boldsymbol{r}_{N-1}-\boldsymbol{r}_N|/c}\right]^\top, \tag{2.13}$$

where $\jmath$ is the imaginary unit and $\omega_k$ is the $k$th angular frequency. For each $\boldsymbol{s}$,

$$P_k(\boldsymbol{s}) = \frac{1}{\boldsymbol{a}_k^{\mathrm{H}}(\boldsymbol{s})\boldsymbol{R}_k\boldsymbol{a}_k(\boldsymbol{s})}, \tag{2.14}$$

is calculated where $^{\mathrm{H}}$ is the Hermitian transpose.

If $\boldsymbol{s}$ is near to the actual source position, $P_k(\boldsymbol{s})$ becomes small. After averaging $P_k(\boldsymbol{s})$ over the target frequency bins ($k_L \leq k \leq k_H$), the source positions can be estimated as

$$\arg\min_{\boldsymbol{s}\in\mathcal{S}} P(\boldsymbol{s}) = \arg\min_{\boldsymbol{s}\in\mathcal{S}} \left(\frac{k_H - k_L}{\sum_{k=k_L}^{k_H} \frac{1}{P_k(\boldsymbol{s})}}\right). \tag{2.15}$$

The M-CSP method outperforms the 2D-CSP and 2D-multiple signal classification (MUSIC) methods[40].

## 2.3 Prior distributions of CSP analysis

This section introduces the prior distribution to CSP analysis to improve the robustness of source localization. When features at the current frame are contaminated by noise, if prior distribution is accurate, accurate estimation can be achieved. Prior distribution of DOA calculated from previous frames is useful to estimate DOA in the current frame. In addition, whether current frame is speech or not is also useful information for the estimation. This information can be obtained by combining source localization with VAD.

Fig. 2.2 shows a schematic diagram of the proposed method. In this section, we define "CSP(I)" and "CSP(II)" as baseline methods; "CSP(I)" is an original CSP analysis and "CSP(II)" is a conventional CSP analysis with a peak-hold process [44] and noise component suppression, which sets the cross power spectrum to zero when the estimated SNR is under 0 dB. This makes CSP more robust for noise and reverberation[2]. The proposed method, "CSP F", has a CSP coefficient filtering process that is added to "CSP(I)" and "CSP(II)" respectively in Section 2.3.1 and 2.3.2.

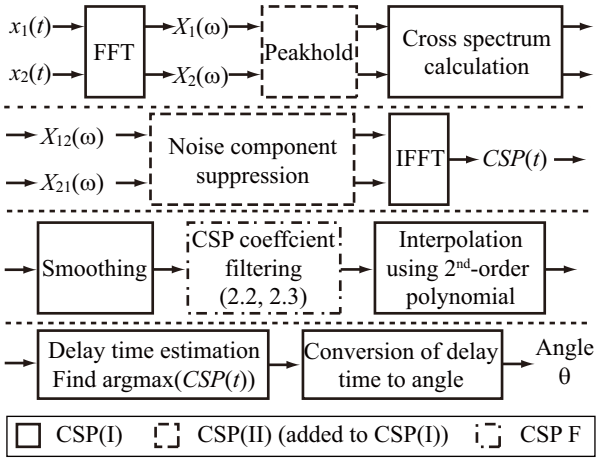



Fig. 2.2 Schematic diagram of the proposed method.

### 2.3.1 Prior distributions of CSP coefficients

We assume that sources do not move substantially and the duration of sources is longer than that of noise. For example, when you operate hands-free devices by voice, the location of speaker would not move substantially. Smoothed CSP coefficient $\overline{CSP}_t(\tau)$ is obtained as in Eq. (2.16) by averaging $CSP_t(\tau)$ during $2d+1$ frames (here, $d=5$).

$$\overline{CSP}_t(\tau) = \frac{1}{2d+1} \sum_{j=i-d}^{i+d} CSP_j(\tau). \tag{2.16}$$

[2]In reverberant environments, robust TDOA estimation [45, 46] methods have been proposed.

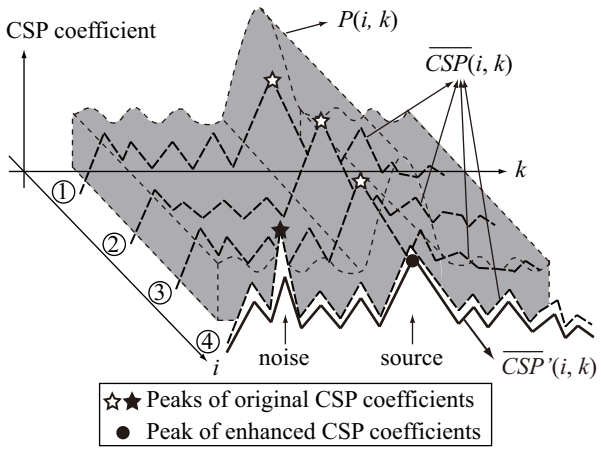



Fig. 2.3 Procedure for calculating a likelihood function of CSP coefficients and eliminating noise.

We assume that $\overline{CSP}_t(\tau)$ is a likelihood of sound source directions, which correspond to a delay time $k$, then we obtain an accumulated likelihood $L_t(\tau)$ as follows:

$$L_t(\tau) = \sum_{j=0}^{i+d} CSP_j(\tau). \tag{2.17}$$

Prior distribution $P_t(\tau)$ $(0 \leq P_t(\tau) \leq 1)$ is normalized by the maximum value of $L_t(\tau)$, as in Eq. (2.18):

$$P_t(\tau) = \frac{\max\left(L_t(\tau), 0\right)}{\max\left(L_t(0), L_t(1), \cdots, L_t(\tau_{max})\right)}, \tag{2.18}$$

where max is a function that returns a maximum value of arguments. Finally, the filtered CSP coefficient $\overline{CSP}'_t(\tau)$ is obtained by combining a weighted CSP coefficient whose weight is $P_t(\tau)$ with an original coefficient at the combination ratio $r$, as in Eq. (2.19):

$$\overline{CSP}'_t(\tau) = \left(r + (1-r)P_t(\tau)\right)\overline{CSP}_t(\tau). \tag{2.19}$$

If source does not move and SNR is high, simple averaging as in Eq. (2.20) is effective.

$$\overline{CSP}''_t(\tau) = \frac{\sum_{j=0}^{i+d} CSP_j(\tau)}{i+d+1}. \tag{2.20}$$

However, if large CSP value of noise inputs, estimation accuracy decreases significantly because the peak of noise hardly diminishes by averaging. The proposed method only suppresses the noise component and does not increase the peak value of CSP coefficient because $P_t(\tau)$ is 1 or less. Hence, the proposed method is less affected by noise than the simple averaging.

Fig. 2.3 clearly shows the above-mentioned procedure. We obtain $\overline{CSP}_t(\tau)$. The source direction is the center, but a noise peak appears on the left side in the $4^{\text{th}}$-frame. This peak is indicated as a closed star in Fig. 2.3. If the source does not move substantially, the center peak

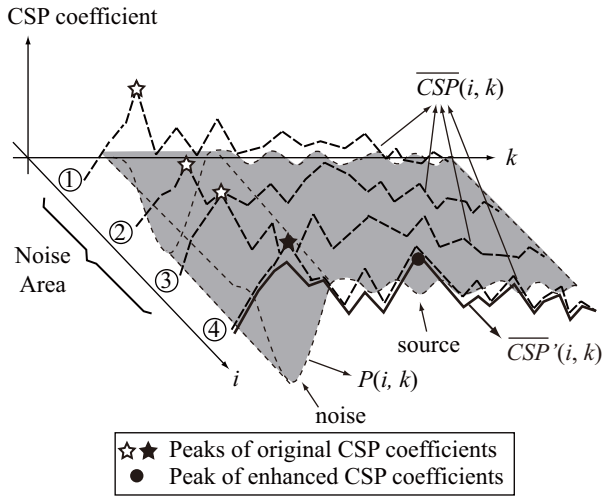



Fig. 2.4 Procedure for calculating a likelihood function of CSP coefficients with VAD information and eliminating the disturbance of noise.

is reliable because the difference between the noise peak and the center peak is small. Then, we calculate $P_t(\tau)$ using accumulated CSP coefficients, as in Eq. (2.18). The central section of a likelihood function is greater than other sections, because previous peaks have appeared at the center. We localize the source at the center, indicated by a closed circle, by multiplying $P_t(\tau)$ and $\overline{CSP}_t(\tau)$ and combining these and the original CSP coefficients at the ratio $r$, as in Eq. (2.19).

### 2.3.2 Combination with voice activity detection information

If the target is speech, peaks of CSP coefficients in non-speech areas are attributed to noise. A modified likelihood $L'(i, k)$ is obtained as in Eq. (2.21) according to VAD information.

$$L'_t(\tau) = \sum_{j=0}^{i+d} ((1+\alpha)\delta(j) - \alpha)\, CSP_j(\tau), \tag{2.21}$$

where $\delta(j)$ is a function that returns unity in speech areas and zero in non-speech areas at the $j$th frame, and $\alpha$ ($\alpha > 0$) is a penalty. This leads to a sign inversion of CSP coefficients in non-speech areas. Peaks of speech are enhanced by suppressing noise peaks, which are dominant in non-speech areas. Fig. 2.4 illustrates this procedure. The first three frames are noise according to VAD. The speech peak appears at the center in the $4^{\text{th}}$-frame. However, the noise peak on the left side indicated as a closed star is higher. In this case, likelihood $L'_t(\tau)$ is obtained as in Eq. (2.21) by inverting any sign of CSP coefficients in the noise area and $P_t(\tau)$ is estimated from these likelihoods. The speech peak is enhanced and a source is located as a closed circle using the same procedure as in Eq. (2.19).

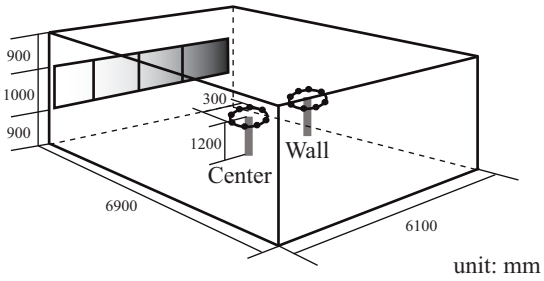



Fig. 2.5 Geometry of a meeting room.

Table 2.1 Recorded noise that is added to evaluation data. (SNR of 6 and 24 dB)

| Name | Explanation |
|---|---|
| AirCon | Air Conditioner noise. |
| 180F (270F) | Stamping at 180° (270°). |
| OpenWin | Environmental noise with windows open. |
| RotateF | Rotating around the microphone array with stamping. |

### 2.3.3 Experimental setups

To validate the effectiveness of the proposed method, source localization experiments under noisy and reverberant environments were performed. Simulated data were made by clean speech, impulse responses, and noises. Impulse responses were measured at every 30° at the center and near the wall of the room as shown in Fig. 2.5. The minimum and maximum distances between sources and the center of a microphone array (8 ch circle array) $\rho$ were 1 and 2 m, respectively. Reverberation time $T_{30}$ was 0.68 s and the reverberation decay curve was not bent. Evaluation data was produced by convolving speech (control words for air conditioner) with these impulse responses. Speaker direction was located using two diagonal microphones. Recorded noise (Table 2.1) was added to the evaluation data at SNR of 6 and 24 dB. Sampling frequency $f_s$ was 16 kHz, and the window length and frame shift of STFT were 60 ms and 30 ms, respectively. According to preparatory studies, $r$ was 0.3 [Eq. (2.19)] and $\alpha$ was 1.0 [Eq. (2.21)]. Note that performance depended little on these parameters.

### 2.3.4 Results and discussion

#### 2.3.4.1 Comparison with the two-microphone method

Estimation accuracy in the speech area is calculated by each frame when the microphone array is located at the center of the room. The error tolerance is ±15°. The estimation average accuracy in all directions is shown in Figs. 2.6 and 2.7, which represent the easiest case (SNR of 24 dB, $\rho$ of 1 m) and the most difficult case (SNR of 6 dB, $\rho$ of 2 m), respectively. A combination of "CSP(I)" and the proposed method ("CSP F") improves the accuracy in Fig. 2.6 but not in

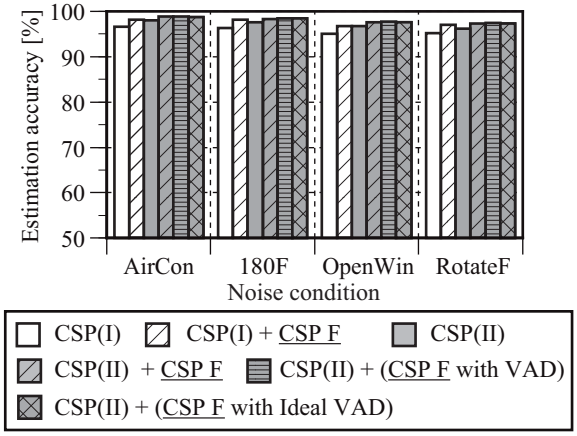



Fig. 2.6 Estimation accuracy of arrival direction when SNR is 24 dB and the distance between the source and the center of the microphone array $\rho$ is 1 m.

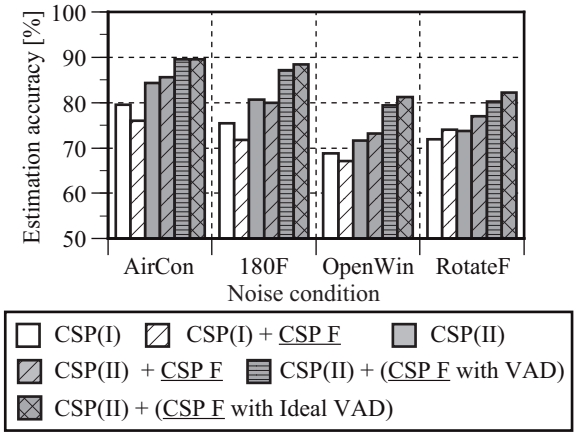



Fig. 2.7 Estimation accuracy (SNR of 6 dB, $\rho$ of 2 m).

Fig. 2.7, because of less accurate prior distributions when SNR is low. Accuracy increases with a combination of "CSP(II)" and "CSP F", because the noise components of the cross power spectrum are reduced in advance. In addition, the use of VAD [47] improves the performance of "CSP F" because the estimation accuracy of prior distributions increases as noises are learned ("CSP F with VAD"). The difference in accuracy between automatic VAD and manually tagged VAD ("CSP F with Ideal VAD") is not large.

Fig. 2.8 shows CSP coefficients on a time-angle plane with directional noise at "180F" and speech at 60° (SNR of 6 dB, $\rho$ of 2 m). Using the proposed method, maximal peaks created by foot noises near 180° are suppressed, maintaining the speaker peaks near 60°, as seen on the right graph, unlike that observed for the peaks of the conventional method, as seen on the left graph. Fig. 2.9 shows the section at A-B (1.02 s) of Fig. 2.8. Originally, the peak appears close to 160°, but with the proposed method, it is correctly estimated closed to 60°.

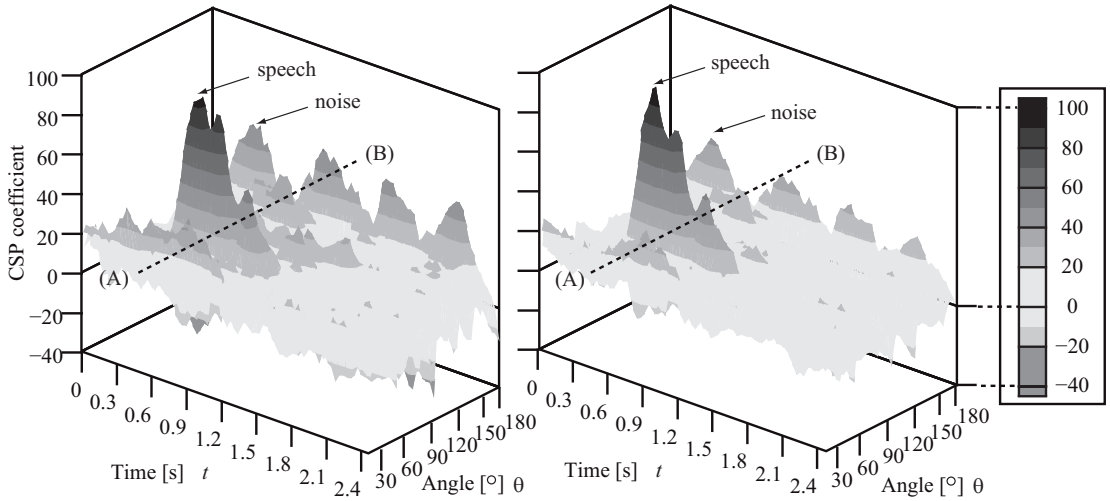



Fig. 2.8 Original and enhanced CSP coefficients on the time-angle plane. (left: original, right: enhanced)

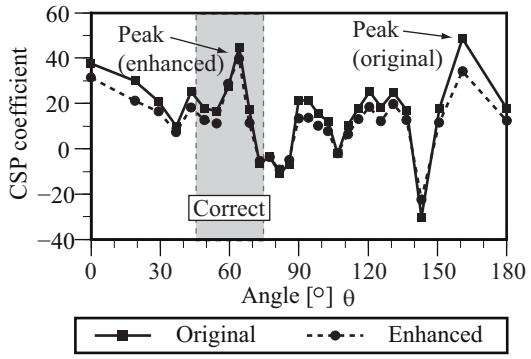



Fig. 2.9 Enhanced CSP coefficient ($t = 1.02$ [s]).

#### 2.3.4.2 Comparison with four- and eight-microphone methods

Using three or more microphones reduces noise by synchronously adding paired CSP coefficients [35]. We compared the proposed method with these types methods that use three and four microphones. Fig. 2.10 shows the results using four microphones (all six pairs) and eight microphones (four diagonal pairs). The accuracy of the proposed method is superior to the method that uses four microphones and is equivalent to the method that uses eight microphones. Note that multiple microphone methods require additional computational costs, but the proposed method do not require it.

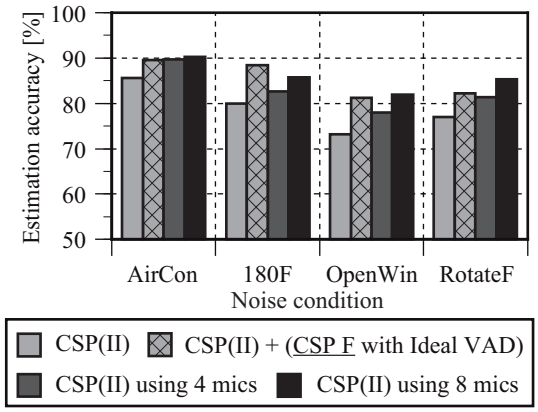



Fig. 2.10 Estimation accuracy of the proposed method (SNR of 6 dB, $\rho$ of 2 m) compared with that using four and eight microphones.

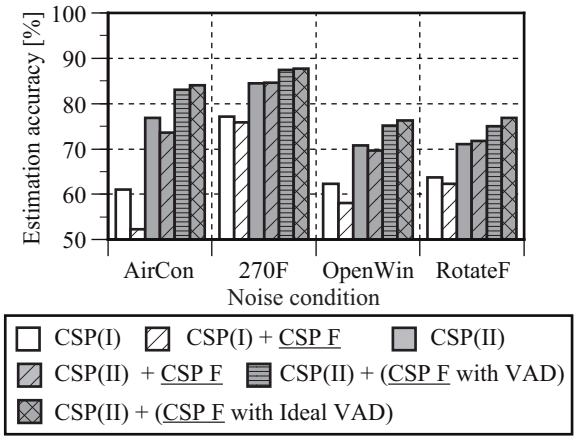



Fig. 2.11 Estimation accuracy (SNR of 6 dB, $\rho$ of 2 m). Sources and receivers are located near the wall.

#### 2.3.4.3 Effect of receiving point

Fig. 2.11 shows the accuracy when the microphone array is near the wall (SNR of 6 dB, $\rho$ of 2 m). Although it is difficult to estimate DOA because the first order reflected sounds are stronger than the center case, the estimation accuracy is improved by using a combination of "CSP(II)" and "CSP F with VAD".

## 2.4 Template-based method for compensation of time difference of arrival (TDOA)

### 2.4.1 Generalized cost function of source localization

In real situations with reverberation, the theoretical TDOA and the observed TDOA for the correct positions can differ due to reverberation or measurement errors. To reduce their influence, we propose a template-based method whose cost function $P$ is given in a generalized form as

$$\arg\min_{\boldsymbol{s}\in\mathcal{S}} P(\boldsymbol{s}) = \arg\min_{\boldsymbol{s}\in\mathcal{S}} \sum_{m=1}^{M} \left(\tau^{ref}_{\varphi(m)}(\boldsymbol{s}) - \tau^{obs}_{\varphi(m)}\right)^2, \tag{2.22}$$

where $\tau^{ref}_{\varphi(m)}(\boldsymbol{s})$ is the reference TDOA for position $\boldsymbol{s}$ and $\tau^{obs}_{\varphi(m)}$ is the observed TDOA. The 2D-CSP method uses $\tau^{sph}_{\varphi(m)}$ for reference and $\tau^{csp}_{\varphi(m)}$ for observation[3].

### 2.4.2 Template that modifies reference TDOA

Source localization methods use the theoretical TDOA as a reference, but observations generally contain errors. For example, reflected waves have high correlations with direct waves, which leads to TDOA estimation errors. Fig. 2.12 shows the errors caused by reverberation. In this case, the observed TDOA $\tau^{csp}_{1,2}$ is longer than the theoretical one $\tau^{csp}_{1,2}$ when the direct wave for the first microphone and the reflected wave for the second microphone have higher correlations than direct waves. The errors between the theoretical TDOA and observed TDOA are denoted as $\epsilon$. The reference TDOA is modified by the errors $\epsilon$, which are calculated for known positions $\boldsymbol{s} \in \mathcal{S}$ in the reference measurements a priori after $\tau^{obs}$ is calculated as Eq. (2.23).

$$\epsilon_{\varphi(m)}(\boldsymbol{s}) \leftarrow \tau^{obs}_{\varphi(m)} - \tau^{sph}_{\varphi(m)}(\boldsymbol{s}). \tag{2.23}$$

In the 2D-CSP case, this formula is

$$\epsilon_{\varphi(m)}(\boldsymbol{s}) \leftarrow \tau^{csp}_{\varphi(m)} - \tau^{sph}_{\varphi(m)}(\boldsymbol{s}). \tag{2.24}$$

Prior to the first use, these errors $\epsilon$s are calculated for the target points $\mathcal{S}$ and stored as a template. These modified references are expected to cancel out the errors. In the case without errors or reflections, $\epsilon$ is zero and the proposed method exactly matches the original method.

By considering the stored $\epsilon$ for position $\boldsymbol{s}$, we use the reference below, $\tau^{ref}_{\varphi(m)}$, instead of $\tau^{sph}$:

$$\tau^{ref}_{\varphi(m)}(\boldsymbol{s}) \approx \tau^{sph}_{\varphi(m)}(\boldsymbol{s}) + \epsilon_{\varphi(m)}(\boldsymbol{s}). \tag{2.25}$$

This method can be applied to any source localization method, no only the 2D-CSP method.

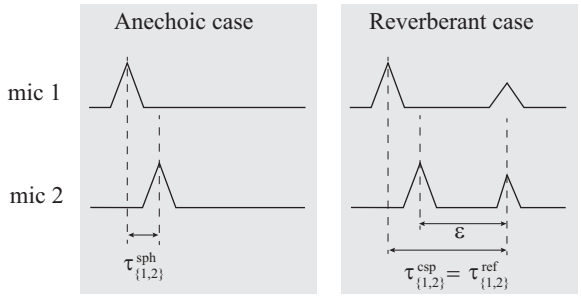



Fig. 2.12 TDOA errors caused by reverberation.

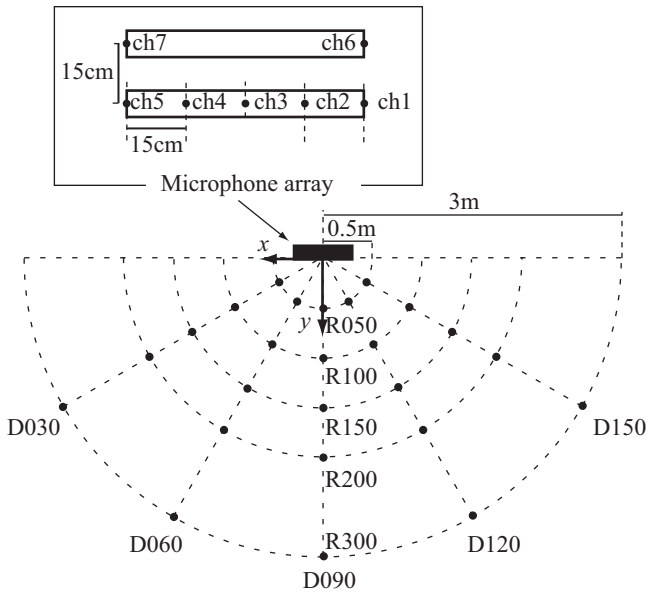



Fig. 2.13 Setting of source points and microphone array.

### 2.4.3 Experimental setups

Experiments validated the effectiveness of the proposed template-based method. The number of sound sources was 25 in the room shown in Fig.2.13 and 25 corresponding impulse responses were recorded. The notation D{$D$}R{$R$} denotes that the source exists in direction $D$ [°] and at a distance $R$ [cm] from the center of the microphone array, which is the origin. To construct evaluation data, the impulse responses were convolved with clean speech utterances, which were composed of a few words each. Vocabularies were control words for air conditioners, such as "Set to 30 degrees.". The room reverberation time measured at the room center was 580 ms, where there were rich reflections in this room. We validated two cases: a reverberant condition in Sect. 2.4.4.1 and a reverberant and noisy condition, where the air conditioner noise was added

[3]We employ 2D-CSP and M-CSP methods as a baseline, but our proposed template-based method can be applied to any methods. This paper assumes time-invariant systems.

at a signal-to-noise ratio (SNR) of 12 dB, in Sect. 2.4.4.2. The sampling frequency was 16 kHz[4], and the window length and frame shift of the short-time Fourier transform were 60 ms and 30 ms, respectively. Frequency bands higher than 150 Hz were used. The candidate points of the sound source were 25 positions, which were the same as the above source positions. Seven microphones were prepared for the recordings as shown in Fig.2.13. In this case, the sound speed $c$ and the microphone positions $\boldsymbol{r}_i$ are constant and known. We show the results using three microphones (chs. 1, 2 and 5) (abbreviated to pair-3ch) and five microphones (chs. 1, 3, 5, 6 and 7) (abbreviated to pair-5ch). The microphone pairs were all the pairs for all the methods ($M = {}_N C_2$). For our proposed method, the reference measurements were 10 utterances at each point by one female speaker, who was different from the evaluation speakers.

The estimation performances were evaluated on the basis of the distance between the actual source position $\boldsymbol{s}_a$ and the estimated source positions $\boldsymbol{s}_e$ in terms of two measures: the estimation accuracy within 25% tolerance (shown in a bar graph) [%] (i.e., the ratio of the number of cases where $|\boldsymbol{s}_a - \boldsymbol{s}_e|/|\boldsymbol{s}_a|$ is less than 0.25 to the total number of cases) and the average absolute error (shown in a line graph) [m] $|\boldsymbol{s}_a - \boldsymbol{s}_e|$. Two types of evaluation are necessary because, for the former one, farther sources have larger tolerance errors, whereas for the latter one, nearer sources have larger tolerance errors relative to the actual distance. Tolerance errors are different for different applications but 30 cm errors can be tolerated for many applications. One of the examples is home appliances such as air conditioners that detect humans and concentrate the air flow to that area. In this kind of application, 30 cm errors are acceptable.

### 2.4.4 Results and discussion

#### 2.4.4.1 Reverberant condition

First, we compared two conventional methods: the 2D-CSP and M-CSP methods. Fig. 2.14 shows the average estimation accuracies and errors. The performance of the M-CSP method was higher than that of the 2D-CSP method as shown in[40]. Fig. 2.15 shows contours of the estimation accuracies of the 2D-CSP method, which show that there were many points that could not be estimated. Compared with the 2D-CSP method, the M-CSP method improved the estimation accuracies on average; however, Fig.2.16 shows that there were still many points that could not be estimated.

To evaluate the tolerance errors of direction estimation and source localization, Fig.2.17 shows the value of the cost function $P$ (pair-5ch). The differences in $P$ between different distances in the same direction were much smaller than those between directions for the same distance. This shows the difficulty of 2D-source localization.

Fig. 2.14 also shows the result of our method. We validated the effectiveness of our '2D-CSP+template' method. Fig. 2.18 shows that in pair-3ch, there were some points that had low

[4]This is the most widely used sampling frequency. If the sampling frequency is higher, the estimation performance can be improved. In such a scenario, the tendencies are the same, although the baseline of the conventional method is also higher and the proposed method can improve the performance further.

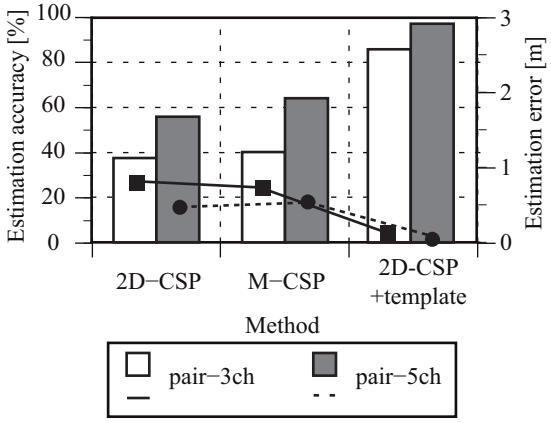



Fig. 2.14 Average estimation accuracy [%] with ±25% tolerance (bar graph) and estimation error [m] (line graph) under reverberant condition.

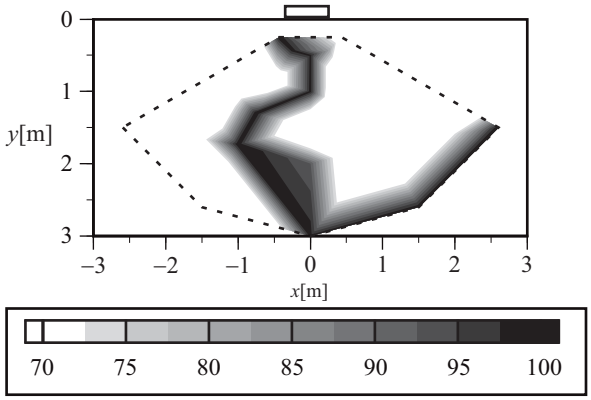



Fig. 2.15 Contours of estimation accuracy [%] of the 2D-CSP method with ±25% tolerance using pair-5ch.

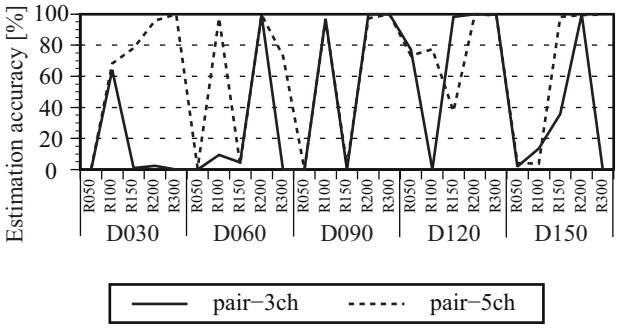



Fig. 2.16 Estimation accuracy [%] of the M-CSP method with ±25% tolerance at each point.

accuracies, but, in pair-5ch, almost all the points had high accuracies of over 90%. Calibrations effectively reduced the influence of reflected sounds and measurement errors.

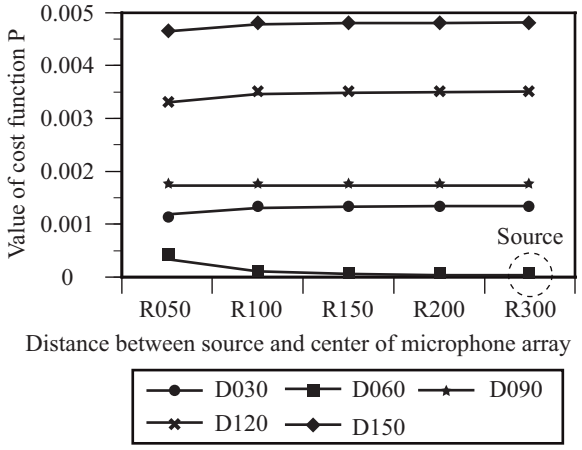



Fig. 2.17 Value of cost function $P$ in Eq. (2.10): The source is located at "D060R300".

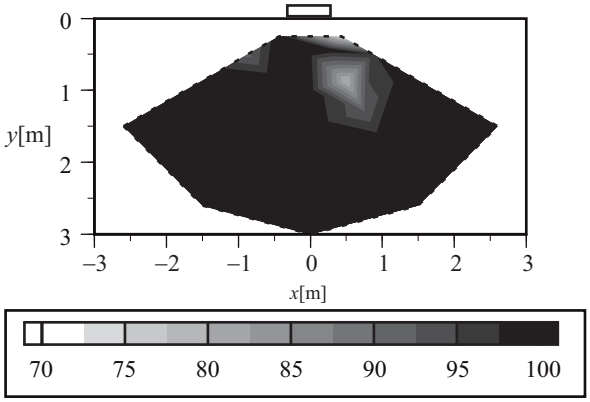



Fig. 2.18 Contours of estimation accuracy [%] of the 2D-CSP+template method with ±25% tolerance using pair-5ch under reverberant condition.

Fig. 2.19 shows a comparison of the computational times of the above methods. All the methods were implemented by C++ and computational times were obtained using the same computer. Computational times were normalized by that of the 2D-CSP method. The computational time of the M-CSP method was 30 times longer than those of the 2D-CSP and 2D-CSP+template methods. Those of the pair-5ch case were two or three times larger than those of the pair-3ch case, which were approximately proportional to the number of pairs from 3 to 10.

#### 2.4.4.2 Reverberant and noisy condition

Fig. 2.20 shows the accuracies under a reverberant and noisy environment. By comparison with Fig.2.14, it can be seen that noise degraded the estimation performance. The 2D-CSP method had unsatisfactory performance but the 2D-CSP+template method improved the performance. The proposed method improved the performance less than that for the reverberant case because

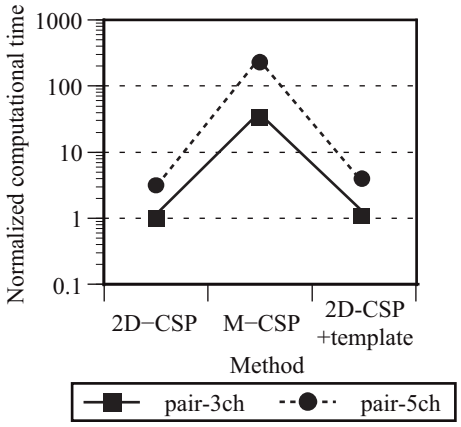



Fig. 2.19 Computational time normalized by that of the pair-3ch case for the 2D-CSP method.

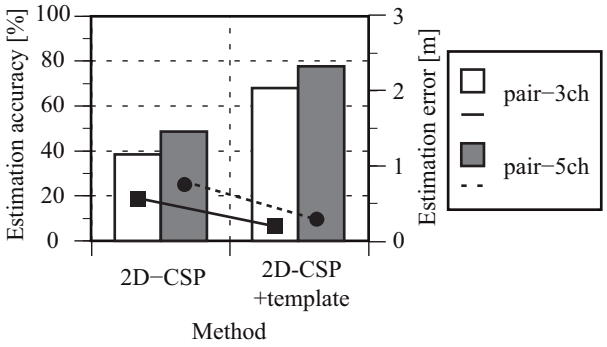



Fig. 2.20 Average estimation accuracy [%] and estimation error [m] with air conditioner noise at an SNR of 12 dB.

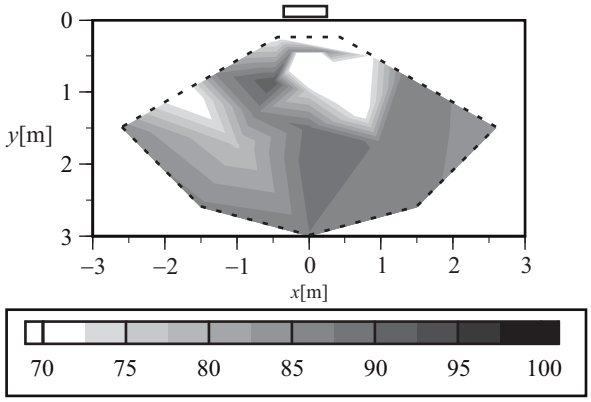



Fig. 2.21 Contours of estimation accuracy [%] of the 2D-CSP+template method with ±25% tolerance using pair-5ch with air conditioner noise at an SNR of 12 dB.

the template was built for the reverberant case (i.e., without noise). Fig. 2.21 shows the contours of the estimation accuracies of the 2D-CSP+template method. For almost all the points, the

accuracy was more than 80%. Even without noise, the conventional method was not accurate at some points as shown in Fig.2.15. The proposed method is much more practical than the conventional method. We thus validated the effectiveness of our 2D-CSP+template method for a reverberant and noisy condition.

## 2.5 Conclusion of the chapter

This chapter first overviews conventional localization techniques. Among them, CSP analysis is effective for source localization but its performance degrades due to noise and reverberation. First, we propose a method that reduces the effect of noise for the estimation of DOA by CSP analysis. The proposed method uses prior distributions estimated from accumulated CSP coefficients. We demonstrated that this method was effective for both diffusive and directional noise and that using VAD information improved estimation accuracy.

Second, we proposed a template-based method in order to reduce the influence of reflected sounds and measurement errors further. Without increasing the computational time, our method can improve source localization accuracies for reverberant and noisy environments.

Journal papers related to this chapter are [37, 42].

# 3 Front-end techniques for robust automatic speech recognition (ASR)

## 3.1 Introduction

Front-end techniques are essential for robust ASR even for high performance ASR systems such as DNN-based ones. Fig. 3.1 shows the front-end techniques, which are described in this section. There are three elements: SE, VAD, and mismatched condition compensation, In the case of noisy speech ASR, noise reduction is necessary. In addition, it is necessary to address reverberation, which is composed of reflected sounds from walls, ceilings, or furniture, in addition to the direct sound from a sound source. Reverberation as well as noise degrades the intelligibility of speech for humans, and it also significantly degrades ASR performance. Thus, two types of speech enhancement are necessary: noise reduction and dereverberation. Table 3.1 shows that various SE methods are classified into single-channel and multi-channel ones. In general, multi-channel ones outperform single-channel one because multi-channel techniques can use much richer information, especially spatial information, than single-channel ones. Section 3.2 proposes a single-channel dereverberation method that can automatically estimate reverberation time. Reverberation time is an important parameter that represents the extent of reverberation in a target room. Section 3.3 proposes a multi-channel noise reduction method that combines BM and IVA. Section 3.4 proposes an initial value setting method for MNMF, which is a multi-channel noise reduction method.

VAD is an also essential technique for ASR. If noise can completely reduced, it is meaningless that speech activation cannot be detected. Section 3.5 proposes a VAD technique whose models are trained by density ratio estimation.

Mismatch compensation due to inappropriate settings is another essential technique. There are many mismatches between training and evaluation. These mismatches degrade ASR perfor-

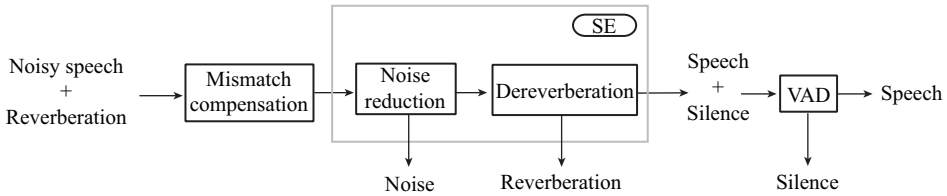



Fig. 3.1 Front-end process appeared in Chapter 3.

Table 3.1 Classification of various speech enhancement techniques. (NMF: non-negative matrix factorization, SS: spectral subtraction, WF: Winner filter, ILRMA: independent low-rank matrix analysis, MINT: multiple-input/multiple-output inverse filtering theorem, BM: binary masking, ICA: independent component analysis, IVA: independent vector analysis)

| target | noise | reverberation |
|---|---|---|
| single channel | NMF [48, 49]<br>SS [52], WF | Multi-step linear prediction [50, 51]<br>SS-based method (3.2)<br>the use of harmonic structure [53, 54] |
| multi channel | MNMF (3.4.1), ILRMA [55]<br>BM (3.4.4.1)<br>ICA [58], IVA (3.3.2) | MINT [56]<br>Subspace method [57] |

mance. Section 3.6 investigates the influence of clipping on ASR performance. Clipping is caused by inappropriate recording levels. Section 3.7 investigates the influence of sampling frequency mismatch between training and evaluation.

## 3.2 Single-channel spectral-subtraction-based dereverberation

This section focuses on single-channel dereverberation because these types of methods based on a statistical model of reverberation[59, 60] need relativelly low computational costs and are robust. Lebart *et al.* proposed a dereverberation method [61] using Polack's statistical model [60], whose parameter is reverberation time (RT). This method is effective and its computational load is relatively low; however, its performance is unstable because it estimates RT only from the end of an utterance. Gomez *et al.* proposed an effective method of the dereverberation of late reverberation, but this method requires an impulse response in a room to have been measured in advance [62]. Löllmann *et al.* also used a statistical model whose RT is estimated by a maximum likelihood approach [63]. This method needs more parameters and computational load than Lebart's method. The key to using statistical models for dereverberation is to limit the number of parameters and to estimate them robustly.

This section proposes a dereverberation method in which SS is used [52]. We also use Polack's statistical model and propose a method of estimating RT. In [61], RT is estimated only from the end of utterances, which is inappropriate for speech recognition because RT must be estimated in a short time. It is also difficult to detect the end of an utterance robustly, considering the overlap of utterances. On the other hand, because speech has sparseness in the time-frequency domain [64], we can utilize the decay characteristic of not only the end of utterances but also whole utterances at the frequency bin. Concretely, the proposed method estimates RT from floored ratios after SS. The floored ratio is the ratio of the number of floored points by SS to

the total number of points on the time-frequency plane. This is a more robust and effective algorithm than [61]. Additionally, the proposed algorithm does not require training data and reverberation characteristics in a room unlike [62], and is much simpler than [63]. First, we clarify the relationship between the subtraction coefficients of SS and RT. Second, we propose an algorithm for estimating RT utilizing the relationship between the floored ratio and RT. Experiments show that RT can be estimated from observed speech, and speech is robustly dereverberated in unknown environments at a low computational load.

In reverberant environments, observed sounds are smeared by reflected sounds and modeled as

$$x(s) = \sum_{\nu=0}^{\infty} h(\nu) y(s-\nu), \tag{3.1}$$

where $x$, $y$, $\nu$, and $h$ are the observed sound, the source sound at the current sample number $s$, the delay sample, and the impulse response, respectively.

If RT is much longer than the frame size of the STFT, the energies of the reflected and direct sounds can be simply superposed because they are incoherent [65]. Therefore, an observed power spectrum $|X(t,k)|^2$ is modeled as a weighted sum of the source's power spectrum $|Y(t,k)|^2$ as

$$|X(t,k)|^2 \approx \sum_{\mu=0}^{i} w(\mu) \cdot |Y(t-\mu,k)|^2, \tag{3.2}$$

where $t$, $k$, $\mu$, and $w(\mu)$ are the current frame index, the index of an $K$-dimensional STFT bin ($1 \leq k \leq K$), the delay frame, and the weight coefficient ($0 \leq \mu \leq t$), respectively. Although [61] estimates the reverberation power spectrum by calculating the running average of the observed power spectrum and delaying it for 50 ms, the proposed method considers all past observed sounds.

### 3.2.1 Relationship between subtraction coefficients and reverberation time

We assume that a source's power spectrum is calculated from an observed power spectrum using an energy-increasing ratio $q(T_r)$ caused by reverberation as

$$|Y(t-\mu,k)|^2 \approx \frac{1}{q(T_r)} |X(t-\mu,k)|^2, \tag{3.3}$$

where $T_r$ is RT in the evaluation environment and $q(T_r)$ is an increasing function of $T_r$ because the longer $T_r$ is, the more reverberant components are added to $|X(t,k)|^2$. Assuming that $w(0)$ is unity, we derive Eq. (3.4) from Eqs. (3.2) and (3.3).

$$|Y'(t,k)|^2 = |X(t,k)|^2 - \sum_{\mu=1}^{i} w(\mu) \cdot |X(t-\mu,k)|^2 \tag{3.4}$$

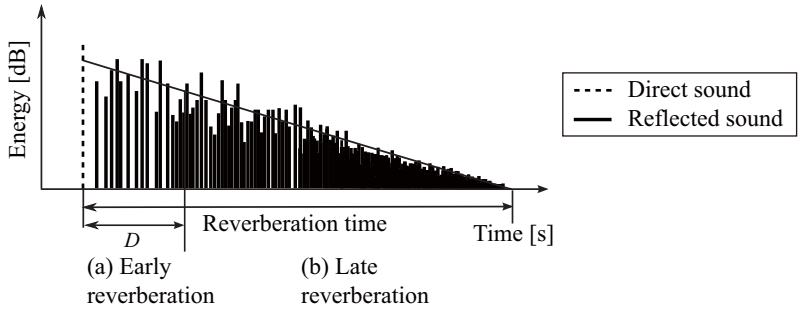



Fig. 3.2 Early and late reverberation. Early reverberation has complex and sparse reflections. Late reverberation has dense reflections and an exponentially decayed shape.

Here, $|Y'(t,k)|^2$ is an estimated dereverberated power spectrum. Once $w(\mu)$ is determined, we dereverberate speech using SS, because the right-hand side of Eq. (3.4) is known and the left-hand side is the result of SS. Although the absolute value of the spectrum was dereverberated in [61], it is natural to dereverberate the power spectrum because $w(\mu)$ is defined in the energy domain, as in Eq. (3.6).

Reverberation is divided into two stages, as shown in Fig. 3.2. Duration $D$ is the threshold between (a)early reverberation with sparse reflected sounds and (b)late reverberation with dense reflected sounds. Because late reverberation mainly degrades the ASR performance [66], in stage (a), dereverberation is not necessarily required, whereas, in stage (b), dereverberation is required. Sound-energy density decays exponentially with $\tau$[s] according to Polack's statistical model [60], and the spatial average of sound-energy density $\bar{E}(\tau)$ is represented by

$$\bar{E}(\tau) = \bar{E}(0)e^{-2\Delta\tau}, \tag{3.5}$$

where $\Delta$ is $\frac{3\ln 10}{T_r}$. Hence, $w(\mu)$ is determined as

$$w(\mu) = \begin{cases} 0 & (1 \le \mu \le D)\,, \\ \frac{\alpha}{q(T_r)} e^{-2\Delta\frac{\varphi}{f_s}\mu} & (D < \mu)\,, \end{cases} \tag{3.6}$$

where $\varphi$ is the frame shift. Similarly to SS, we must set a subtraction parameter $\alpha$ ($> 0$) and a flooring parameter $\beta$ ($0 \le \beta < 1$). Here, we set $\alpha/q(T_r)$ and $\beta$ as 5 and 0.05, respectively. If the subtracted power spectrum is less than $\beta|X(t,k)|^2$, it is substituted by $\beta|X(t,k)|^2$. This is called a flooring process.

$$|Y'(t,k)|^2 = \begin{cases} \beta|X(t,k)|^2 & \text{(floored)} \\ |X(t,k)|^2 - \sum_{\mu=1}^{i} w(\mu)\cdot|X(t-\mu,k)|^2 & \text{(otherwise)} \end{cases} \tag{3.7}$$

### 3.2.2 Estimation of reverberation time

In general, $T_r$ is unknown and must be estimated[1]. We propose a method of estimating $T_r$ from floored ratios, referring to the relationship between the floored ratio and $T_r$. Fig. 3.3 illustrates

[1]In the field of architectural acoustics, some RT estimation methods have been proposed such as [67]

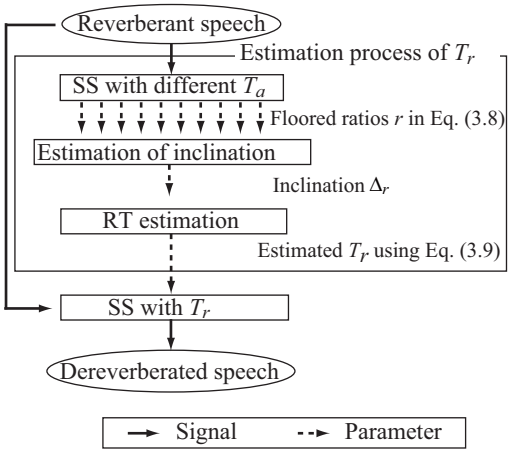



Fig. 3.3 Schematic diagram of the proposed method.

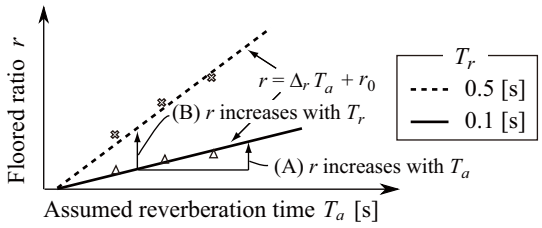



Fig. 3.4 Relationship between some assumed RTs ($T_a$) and floored ratios $r$, with actual RT ($T_r$).

a schematic of the proposed method.

We assume some different RTs ($T_a$) (e.g., we assume 26 $T_a$ from 0.25 to 1.0 s at 0.05 s intervals) and substitute $T_a$ into $T_r$ in Eq.(3.6) for dereverberation. Then we calculate the floored ratio $r$ for each RT. We define the floored ratio $r$ as the ratio of the number of floored points $n_f$ in the time-frequency plane to the number of total points $K \times (t_e - t_s + 1)$, as

$$r = \frac{n_f}{K \times (t_e - t_s + 1)}, \tag{3.8}$$

where the utterance lasts from the frame $t_s$ to $t_e$.

Fig. 3.4 shows two observations of $T_a$ and $r$ at different $T_r$, as follows

(1) Fig. 3.4 (A) shows that $r$ increases monotonically with $T_a$ because the number of floored points at long $T_a$ in Fig. 3.5 (i) is greater than that at short $T_a$ in Fig. 3.5 (ii). This relationship is modeled as $r = \Delta_r T_a + r_0$, where $\Delta_r$ indicates the likelihood of being floored, which is calculated by least-squares regression for two or more $r$.

(2) Fig. 3.4 (B) shows that $r$ increases with $T_r$ at the same $T_a$. Function $q(T_r)$ increases with $T_r$, hence the longer $T_r$ is, the smaller $\frac{\alpha}{q(T_r)}$ is. If we assume $\frac{\alpha}{q(T_r)}$ to be a constant $\alpha_0$, the

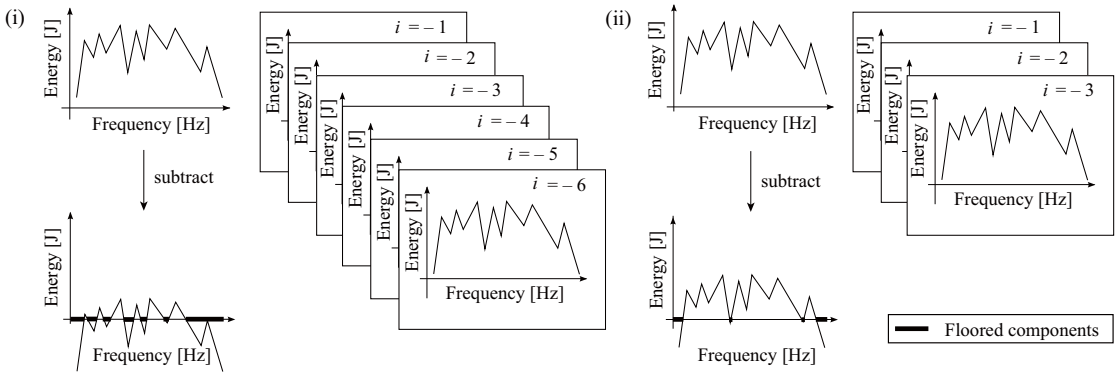



Fig. 3.5 Relationship between an original and dereverberated power spectrum: (i) $T_a$ is long. (ii) $T_a$ is short.

power spectrum after SS is more likely to be floored for a longer $T_r$. For example, if there are three environments with different RTs ($T_r' > T_r'' > T_r'''$) and we assume $\frac{\alpha}{q(T_r)}$ to be $\frac{\alpha}{q(T_r'')}(= \alpha_0)$, oversubtraction is most likely to occur for the environment with $T_r'$ because $\frac{\alpha}{q(T_r')}$ is less than $\alpha_0$. Therefore, $T_r$ has a positive correlation with the likelihood of being floored, $\Delta_r$. We model this relation between $\Delta_r$ and $T_r$ as

$$T_r = \begin{cases} a\Delta_r - b & (a\Delta_r - b > 0), \\ 0 & (a\Delta_r - b \leq 0), \end{cases} \tag{3.9}$$

where $a$ and $b$ are the positive constants and determined heuristically using a test set in advance.

Exploiting these observations, we utilize the inclination $\Delta_r$ of a regression line (from observation 1) to estimate $T_r$ in Eq. (3.9) (from observation 2). The estimation process of $T_r$ is summarized below. This process requires a small amount of computation.

(1) Count the number of floored points $n_f$ after SS on the time-frequency plane at different $T_a$ and calculate the ratio $r$ in Eq. (3.8).

(2) Calculate the inclination $\Delta_r$ of the line $r = \Delta_r T_a + r_0$ by least-squares regression.

(3) Estimate $T_r$ using Eq. (3.9).

### 3.2.3 Experimental setups

To validate the effectiveness of our proposed dereverberation method, we evaluate the word recognition rate using JEIDA-JCSD (B-set) and CENSREC-4 [68]. The former is a data set of 100 area names (e.g., Sapporo) spoken by 20 male and 20 female speakers, and the latter comprises impulse responses recorded in eight real environments with various RTs. Table 3.2 shows RT for each environment determined by the impulse response integration method[69]. The

Table 3.2 Reverberation time $T_r$[s] for each environment determined by impulse response integration method.

| Office | Elevator hall | In car | Living room |
|---|---|---|---|
| 0.22 | 1.16 | 0.09 | 0.77 |
| Lounge | Japanese room | Meeting room | Japanese bath |
| 0.36 | 0.34 | 0.42 | 0.70 |

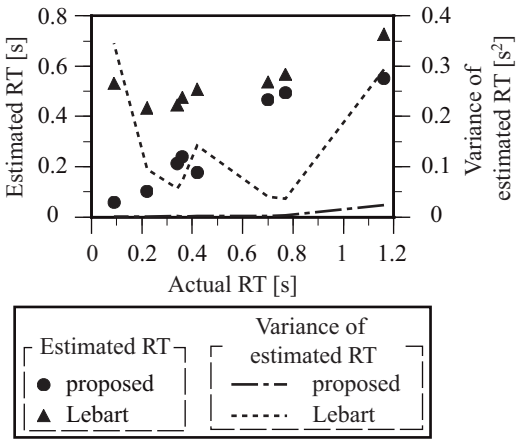



Fig. 3.6 Estimated RTs and their variance, determined by the proposed and Lebart methods with actual RTs.

recognition vocabulary comprised 676 area names, 100 of which were used in evaluations. We used the following parameters: 0-16 order mel-frequency cepstrum coefficients (MFCCs), their $\Delta$ and $\Delta\Delta$, phonemic segment HMMs, and 8-mixture Gaussian distributions. The sampling frequency $f_s$ was 16 kHz and frame size and shift $\varphi$ of STFT were 480 and 160, respectively. Duration of early reflection $D$ is set to 93 ms. We compared the performance of the proposed method with that of Lebart's method[61].

## 3.2.4 Results and discussion

### 3.2.4.1 Estimation accuracy of reverberation time

We evaluate the estimation accuracy of RTs. Fig. 3.6 shows the relationship between the estimated and actual RTs in each environment. We set $a$ and $b$ as 0.0035 and 0.6, respectively, to maximize the recognition rate in a development set. Although these parameters, $a$ and $b$, affect the absolute value of the estimated RT, the magnitude of correlation between each RT always holds independent of these parameters. The correlation coefficient is 0.95, whereas that obtained by the Lebart's method is 0.85. We calculate variances from the results of the estimated RT of the evaluation data (40 speakers×100 utterances). Fig. 3.6 also shows the variance of the estimated

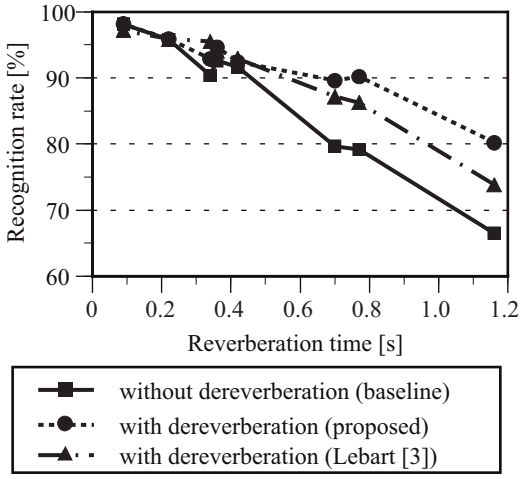



Fig. 3.7 Recognition rate of reverberant speech by the proposed and Lebart methods.

RT. The variance determined by Lebart's method is always larger than that determined by the proposed method, which shows the unstableness of Lebart's method. Because Lebart's method estimates RT from the end of utterances using regression, when RT is short, the regression area is also short and it is difficult to estimate RT. Furthermore, when RT is long, the estimation by regression is difficult for an overlap of utterances or estimation errors.

#### 3.2.4.2 Recognition rate

We evaluate the improvement of ASR performance owing to the incorporation of the proposed method. Fig. 3.7 shows the recognition rate with RT. The proposed method improves the recognition rate for all cases and significantly in three environments ("Japanese bath", "Living room", and "Elevator hall") whose RTs are over 0.5 s. In these three environments, the proposed method improves the recognition rate by 9.9, 11.0, and 13.7%, respectively, whereas those obtaind by Lebart's method are 7.5, 7.1, and 7.3%, respectively. The proposed method improves the average recognition rate by 5.0%, whereas Lebart's method improves that by 3.6%. The recognition rate given by the proposed method is better than that given by the Lebart's method in almost all cases. The proposed method and Lebart's method are equivalent in computational time.

### 3.2.5 Conclusion

In this section, we proposed a dereverberation method with RT estimation. It can yield estimates of RTs and is robust for various environments. The correlation coefficients between the estimated and actual RTs were 0.95. Recognition experiments showed that the recognition

rate was improved in all cases and that the performance was better than that of a conventional method[61] without increasing the computational load.

## 3.3 Combination of binary masking and independent vector analysis (IVA)

The most general SE is based on the physical information such as a DOA of sound sources [70]. This method is fast and effective but susceptible for errors in physical information. On the other hand, blind source separation approach based on statistical independence [71] is more time-consuming and may be inferior to the physical method with precise information but can be robust for measurement errors. This section proposes to combine these physical and statistical approach effectively to improve the robustness of source separations.

### 3.3.1 Binary masking on time-frequency domain

From now on, the number of microphones is two [2]. A cross-spectrum of them is represented as

$$\frac{X_2(t,k)}{X_1(t,k)} = A\mathrm{e}^{\jmath k\tau(t,k)}, \tag{3.10}$$

where $A$ is a positive amplitude ratio, and $\tau(t,k)$ is a time difference between them. Masking matrix $W$ is composed of two vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$:

$$W(t,k) = (\boldsymbol{w}_1(t,k), \boldsymbol{w}_2(t,k))^{\mathrm{H}}. \tag{3.11}$$

If the direction of a sound source $\theta_S$ is known, BM on the time-frequency domain constructs masks $W$ as [72, 73, 70]

$$\boldsymbol{w}_n(t,k) = \begin{cases} \epsilon \boldsymbol{e}_n & (|\frac{c}{l_m}\sin^{-1}\tau(t,k) - \theta_S| > \theta_c), \\ \boldsymbol{e}_n & (\text{otherwise}). \end{cases} \tag{3.12}$$

where $\boldsymbol{e}_n$ is a unit vector whose $n$-th element is one, $\epsilon$ is a small number for smoothing, and $\theta_c$ is a tolerance error; $c$ is a sound velocity and $l_m$ is the distance between microphones. Separated signal $\boldsymbol{Y}$ is obtained as

$$\boldsymbol{Y}(t,k) = W(t,k)\boldsymbol{X}(t,k), \tag{3.13}$$

where $\boldsymbol{X}(t,k)$ and $\boldsymbol{Y}(t,k)$ are vector forms of $[X_1(t,k), X_2(t,k)]^\top$ and $[Y_1(t,k), Y_2(t,k)]^\top$. Separation is effective when physical variables above are all reliable.

### 3.3.2 Independent vector analysis using auxiliary function

Statistical method only assumes an independence between sources and needs no physical information above. The most major statistical method, ICA [58, 74], causes the permutation problem about separated speakers because this method separates sources at each frequency bin [75]. To

[2] This algorithm can be simply applied to the case when the number of microphones is three and more by combining the results of pair-wise maskings

address this problem, IVA minimizes the objective function (3.14) across frequency bins and determines time-invariant separation matrices $W(k)$.

$$J(\boldsymbol{W}) = \sum_k E[r_{n,t}] - \sum_k \ln|\det W(k)|. \tag{3.14}$$

where $\boldsymbol{W}$ is a set of $W(k)$ and $r_{n,t}$ is an auxiliary variable in Eq. (3.15). This can be optimized using an auxiliary function as an upper limit of $J$ [71]. This method outperforms gradient decent based conventional methods. After the update of auxiliary variables as

$$\begin{aligned} r_{n,t} &= \sqrt{\sum_\omega |\boldsymbol{w}_k^{\mathrm{H}}(\omega)\boldsymbol{X}(t,k)|^2}, \\ V_n(k) &= \sum_{t=1}^{T} \left[ \frac{\boldsymbol{X}(t,k)\boldsymbol{X}^{\mathrm{H}}(t,k)}{T r_{n,t}} \right], \end{aligned} \tag{3.15}$$

the separation matrices are updated in two steps: direction update rule

$$\boldsymbol{w}_n(k) \leftarrow \left(W(k)V_n(k)\right)^{-1} \boldsymbol{e}_n, \tag{3.16}$$

and norm normalization rule

$$\boldsymbol{w}_n(k) \leftarrow \frac{\boldsymbol{w}_n(k)}{\sqrt{\boldsymbol{w}_n^{\mathrm{H}}(k)V_n(k)\boldsymbol{w}_n(k)}}. \tag{3.17}$$

Finally, projection back [76] is applied to the separated matrix.

### 3.3.3 Combination of binary masking and IVA

The main reason of degrading physical methods is a spatial aliasing, which occurs in the frequency bands more than $f_c = c/(2l_m)$. For these bands, the performance of physical methods is significantly degraded; on the other hand, statistical method is robust. To address this problem, in the bands less than $f_c$, BM is used and otherwise, IVA is used. However this simple combination causes a permutation problem similar to the ICA, thus we insert BM into the framework of IVA optimization. After BM is applied in the bands less than $f_c$, in the other bands, IVA separates sources where for all $k$'s, auxiliary variables and separation matrices are updated to guarantee the identity of separated speakers. Instead of the update rule (3.16), the update rule follows

$$\boldsymbol{w}_n(k) \leftarrow \begin{cases} \left(W(k)V_n(k)\right)^{-1} \boldsymbol{e}_n & (k > 2\pi f_c), \\ \boldsymbol{e}_n & (\text{otherwise}). \end{cases} \tag{3.18}$$

because, for the frequency bands less than $f_c$, sources are separated by binary masking.

### 3.3.4 Experimental setups

To validate the effectiveness of a combination of binary masking and IVA, experiments on ASR were performed. The impulse responses were measured in a variable reverberant room

Table 3.3 Word accuracy rate [%] in terms of methods and the number of iterations.

| | $l_m = 2.85$[cm] | | | $l_m = 5.7$[cm] | | | $l_m = 37.5$[cm] | | |
|---|---|---|---|---|---|---|---|---|---|
| iter | BM | IVA | prop | BM | IVA | prop | BM | IVA | prop |
| 5 | **84.8** | 61.1 | 84.2 | 76.4 | 60.9 | 78.2 | 37.0 | 57.6 | 56.8 |
| 10 | - | 69.1 | 84.3 | - | 69.3 | 79.1 | - | 64.0 | 61.8 |
| 15 | - | 72.6 | 84.3 | - | 72.5 | 79.0 | - | 66.8 | 64.4 |
| 20 | - | 74.1 | 84.4 | - | 73.5 | **78.9** | - | **68.0** | 65.3 |

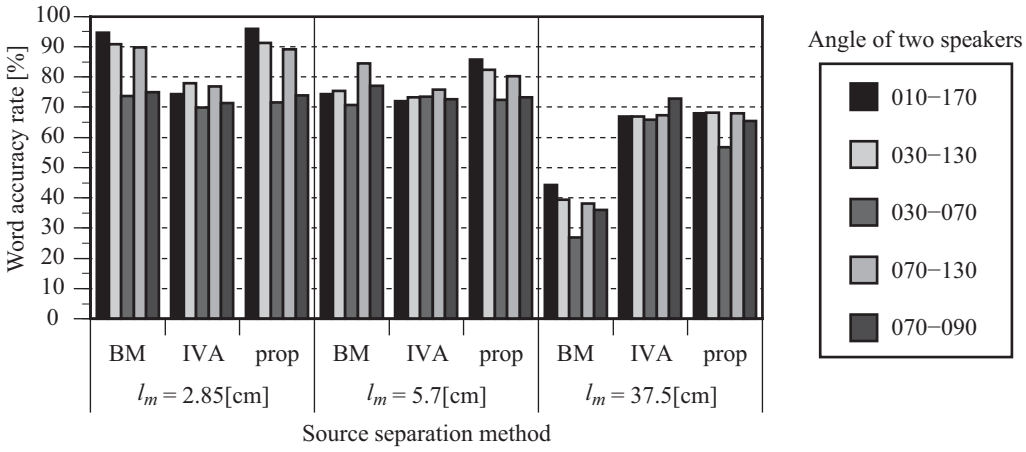



Fig. 3.8 Word accuracy rate [%] in terms of methods (BM: binary masking, IVA: independent vector analysis, and prop: proposed method that combines BM and IVA) and angle of speaker to the microphone array. The iteration number of IVA and prop is 20.

whose reverberation time was 300 ms. This was included in the RWCP database (E2A). Two microphones were picked up from the line array. Microphone intervals $l_m$ were 2.85, 5.7, and 37.5 cm. Direction of arrival was given in this experiment, because that can be estimated at the high accuracy [37]. Impulse responses were provided with the direction of arrival from 10 to 170 degree by 20 degree. This experiment used five combinations of them: (10,170), (30,130), (30,70), (70,130), and (70,90)°. The center of microphone array and a sound source was two meters. Utterances were taken from JEIDA-JCSD (B-set), which was composed of 100 area names. Although the dictionary of the automatic speech recognition system was 100 area names, mixed speech was made from 30 area names with different area names. For speaker variety, twenty speaker sets were prepared from five male and five female speakers. Window length and window shift of STFT were 60 ms and 30 ms, respectively and MFCC features were used.

### 3.3.5 Results and discussion

Table 3.3 shows the relationship between word accuracy rate and the number of iterations for BM, IVA and the proposed method (prop). Note that BM needs no iterations. For IVA and prop, 20 iterations were enough. For the $l_m = 2.85$[cm] case, BM achieved the highest performance, but increasing $l_m$ degraded the performance. IVA was less susceptible for $l_m$, but for the $l_m = 2.85$[cm] case the performance was lower than that of BM. Proposed method achieved the equivalent performance to BM for the $l_m = 2.85$[cm] case and to IVA for the $l_m = 37.5$[cm] case and achieved the best performance for the $l_m = 5.7$[cm] case.

Fig. 3.8 shows the influence of speaker positions. When two speakers are positioned with more than 40 degree intervals, word accuracies were high for BM and prop, and for the $l_m = 2.85$[cm] case, BM and prop achieved the word accuracies more than 90%. IVA was less susceptible for speaker positions.

### 3.3.6 Conclusion

This section proposes a combination of BM and IVA. The former method is based on physical properties of sound propagation and can achieve high accuracy in the situation of little measurement errors but measurement errors such as microphone position discrepancy greatly affect their performance.

The latter method is a statistical method and this can be used for 'blind' situation. These types of methods are robust for the above-mentioned errors, because these methods adjust themselves for these discrepancies.

The combination of them takes advantages of two methods and improves the robustness of source separations. Noisy ASR experiments showed that the proposed method achieved the upper limit performance of two methods.

## 3.4 Coupled initialization of spatial and spectral information for MNMF

One of the most effective SE methods is NMF [48, 49], which factorizes an observation matrix into two matrices: basis and activation matrices. To reconstruct the target signals from the mixed signals, it is important to properly construct the bases. Several methods have been proposed to construct proper initial bases for NMF: the k-means method [77], singular value decomposition [78, 79], and the LBG algorithm [80]. These methods only use training data for basis selection and it is possible that unnecessary bases are included because of a mismatch between training and test data. The small number of bases cannot represent speech well due to the large variety, because spectral properties of speech are dependent on speakers and utterances. The representation capability is improved by increasing the number of bases but it is then difficult to optimize them. Practically, it is necessary to restrict the number of bases and ideally to select bases that fit the phonemes appearing in the utterances in cooperation with ASR. One approach in this direction is ASR-assisted speech enhancement [81, 82, 83, 84], which seems to improve the performance. We extend the approach in [81] to a histogram-based one and validate the effectiveness on an ASR task.

Multi-channel NMF (MNMF) is a multi-channel extension of NMF, which is effective for source separation and noise reduction [85, 86], and factorizes an observation matrix into four matrices. It can consider both spatial and spectral information, simultaneously, by introducing Hermitian semi-positive definite matrices to handle phase information. The separation performance of MNMF is more dependent on initial values than NMF because the number of free parameters is larger.

The introduction of other methods or constraints helps to improve the performance of MNMF. The initial value dependencies are more dominant in the spatial correlation matrix than the other matrices and that its estimation using the cross-spectrum method is effective from enhanced speech by binary masking [87], whereas [88, 55] showed the effectiveness of a rank-1 relaxation. Previous methods initialize bases and spatial correlation matrices, respectively, according to each criterion. However, these are coupled by cluster-indicator latent variables, thus, these spatial and spectral informations should be simultaneously exploited.

This section proposes effective initialization methods for MNMF parameters: ASR-based bases selection (Section 3.4.3), spatial correlation matrix initialization by using the cross-spectrum method and binary masking (Section 3.4.4), and combination of spatial and spectral information by cluster-indicator latent variables initialization (Section 3.4.5). This section validates the effectiveness of the proposed method on the fourth CHiME challenge (details are in Section 5.4), a popular noisy ASR task, and analyzes the influence of each component in terms of the WER.

### 3.4.1 Matrix factorization in MNMF

NMF factorizes an observation matrix $\boldsymbol{X}$ into two matrices: basis matrix $\boldsymbol{T}$ and activation matrix $\boldsymbol{V}$. In addition, MNMF factorizes an observation matrix $\boldsymbol{X}$ into four matrices $\boldsymbol{H}$, $\boldsymbol{Z}$, $\boldsymbol{T}$, and $\boldsymbol{V}$. The two additional matrices $\boldsymbol{H}$ and $\boldsymbol{Z}$ are the spatial correlation matrix and cluster-indicator latent variables, respectively. MNMF clusters $B$ spectral bases into $L$ sources by using the spatial information to achieve high source separation performance without any prior supervised training.

An observation vector is $[X_1, \ldots, X_n, \ldots, X_N]^\top$. The element of an observation matrix $\boldsymbol{X} \in (\mathbb{C}^{N\times N})^{K\times T}$ is represented as

$$\boldsymbol{X}_{kt} = \begin{bmatrix} |x_1|^2 & \cdots & x_1 x_N^* \\ \vdots & \ddots & \vdots \\ x_N x_1^* & \cdots & |x_N|^2 \end{bmatrix}_{kt}, \tag{3.19}$$

where $^*$ denotes the complex conjugate. Matrix $\boldsymbol{X}$ is a hierarchical matrix whose elements $\boldsymbol{X}_{kt}$ are $N \times N$ complex Hermitian positive semi-definite matrices. MNMF factorizes this matrix $\boldsymbol{X}$ into four matrices $\boldsymbol{H}$, $\boldsymbol{Z}$, $\boldsymbol{T}$, and $\boldsymbol{V}$:

$$\boldsymbol{X} \cong \hat{\boldsymbol{X}} = [(\boldsymbol{H}\boldsymbol{Z}) \circ \boldsymbol{T}]\, \boldsymbol{V}, \tag{3.20}$$

where $\circ$ denotes the Hadamard product. Fig. 3.9 illustrates Eq. (3.20); $\boldsymbol{H} \in (\mathbb{C}^{N\times N})^{K\times L}$ is a spatial correlation matrix that indicates the spatial information of $L$ sources and $\boldsymbol{Z} \in \mathbb{R}^{L\times B}$ is a cluster-indicator latent variables matrix that relates spatial information with each basis. Basis matrix $\boldsymbol{T} \in \mathbb{R}^{K\times B}$ is composed of $B$ bases, and $\boldsymbol{V} \in \mathbb{R}^{B\times T}$ comprises the activations of each basis. The right-hand side of Eq. (3.20) can be represented as

$$\hat{\boldsymbol{X}}_{kt} = \sum_b \left[ \sum_l \boldsymbol{H}_{kl} z_{lb} \right] t_{kb} v_{bt}. \tag{3.21}$$

For ideal cases, the reconstructed matrix $\hat{\boldsymbol{X}}$ whose elements are $\hat{\boldsymbol{X}}_{kt}$ matches with the original matrix $\boldsymbol{X}$. However, in general, these matrices differ due to errors. In NMF, an arbitrary distance $D_*(\boldsymbol{X}, \hat{\boldsymbol{X}})$ between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ is defined and the above four matrices in the right-hand side of Eq. (3.20) are updated to minimize this distance. Here, the Itakura-Saito (IS) divergence[3]

$$d_{IS}(\boldsymbol{X}_{kt}, \hat{\boldsymbol{X}}_{kt}) = \mathrm{tr}(\boldsymbol{X}_{kt}\hat{\boldsymbol{X}}_{kt}^{-1}) - \ln\det \boldsymbol{X}_{kt}\hat{\boldsymbol{X}}_{kt}^{-1} - N, \tag{3.22}$$

is used, where $\mathrm{tr}(\cdot)$ is a trace of a matrix.

### 3.4.2 Multiplicative update rule

An iterative optimization algorithm, multiplicative update rule [90], is applied to the randomly initialized non-negative matrices $\boldsymbol{T}$, $\boldsymbol{V}$, and $\boldsymbol{Z}$, and the matrix $\boldsymbol{H}$ whose elements are initialized

[3]IS divergence is suitable for the separation of music and speech, whose dynamic ranges are large [89].

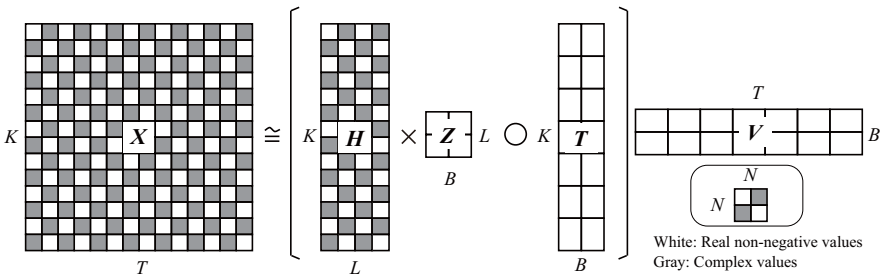



Fig. 3.9 An example of factorizing an observation matrix $\boldsymbol{X}$ into four matrices $\boldsymbol{H}$, $\boldsymbol{Z}$, $\boldsymbol{T}$, and $\boldsymbol{V}$ by the multi-channel NMF algorithm. ($K = T = 7$ and $B = L = N = 2$)

as unit matrices. These matrices are updated to minimize $D_{IS}(\boldsymbol{X}, \hat{\boldsymbol{X}})$ as follows:

$$\begin{aligned}
t_{kb} &\leftarrow t_{kb} \sqrt{\frac{\sum_l z_{lb} \sum_j v_{bt} \operatorname{tr}(\hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{X}_{kt} \hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{H}_{kl})}{\sum_l z_{lb} \sum_j v_{bt} \operatorname{tr}(\hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{H}_{kl})}}, \\
v_{bt} &\leftarrow v_{bt} \sqrt{\frac{\sum_l z_{lb} \sum_i t_{kb} \operatorname{tr}(\hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{X}_{kt} \hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{H}_{kl})}{\sum_l z_{lb} \sum_i t_{kb} \operatorname{tr}(\hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{H}_{kl})}}, \\
z_{lb} &\leftarrow z_{lb} \sqrt{\frac{\sum_i \sum_j t_{kb} v_{bt} \operatorname{tr}(\hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{X}_{kt} \hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{H}_{kl})}{\sum_i \sum_j t_{kb} v_{bt} \operatorname{tr}(\hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{H}_{kl})}}.
\end{aligned} \tag{3.23}$$

$\boldsymbol{H}_{kl}$ is a solution of an algebraic Riccati equation (3.24)

$$\boldsymbol{H}_{kl} \mathsf{A} \boldsymbol{H}_{kl} = \mathsf{B}, \tag{3.24}$$

whose coefficients $\mathsf{A}$ and $\mathsf{B}$ are

$$\begin{cases}
\mathsf{A} = & \sum_k z_{lb} t_{kb} \sum_j v_{bt} \hat{\boldsymbol{X}}_{kt}^{-1}, \\
\mathsf{B} = & \boldsymbol{H}'_{kl} \left[ \sum_k z_{lb} t_{kb} \sum_j v_{bt} \hat{\boldsymbol{X}}_{kt}^{-1} \boldsymbol{X}_{kt} \boldsymbol{X}_{kt}^{-1} \right] \boldsymbol{H}'_{kl},
\end{cases} \tag{3.25}$$

where $\boldsymbol{H}'_{kl}$ is the value of matrix $\boldsymbol{H}_{kl}$ before the update. The solution of Eq. (3.24) is found in the appendix of [86]. It is necessary to normalize matrices $\boldsymbol{H}$ and $\boldsymbol{Z}$, in order to preserve the uniqueness of Eq. (3.20) ($\boldsymbol{H}_{kl} = \boldsymbol{H}_{kl} / \operatorname{tr}(\boldsymbol{H}_{kl})$) and the definition of probabklity ($z_{lb} = z_{lb} / \sum_l z_{lb}$).

Finally, the $l$-th separated source $\tilde{\boldsymbol{y}}_{ktl}$ ($1 \leq l \leq L$) can be obtained by the multi-channel Wiener filter as

$$\tilde{\boldsymbol{y}}_{ktl} = \left[ \sum_b z_{lk} t_{ik} v_{kj} \right] \boldsymbol{H}_{il} \hat{\boldsymbol{X}}_{kt}^{-1} [X_1(t,k), \ldots, X_N(t,k)]^\top \tag{3.26}$$

### 3.4.3 ASR-based initialization of speech bases

Fig. 3.10 shows the selection of speech bases $\boldsymbol{T}$ based on the ASR results. A total of $B$ bases are composed of $B_s$ speech bases and $B_n$ noise bases. The noise bases are randomly initialized in the same manner as the conventional method.

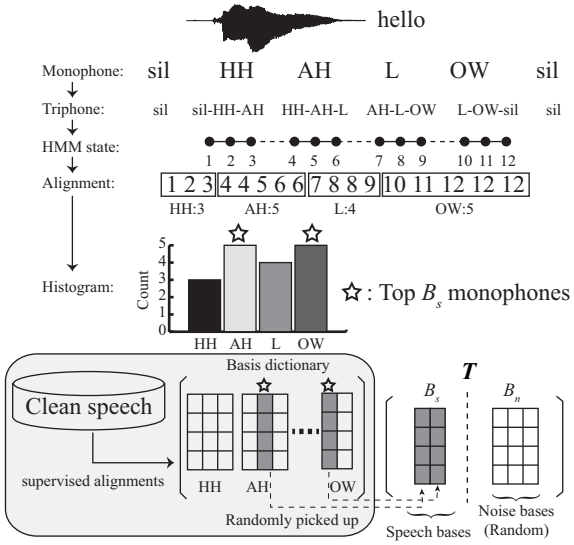



Fig. 3.10 Procedure of the ASR-based speech bases initialization ($B_s = 2$ and $B_n = 3$).

Initial speech bases are sampled from the excerpt of the prepared clean speech. First, a basis dictionary is created from the clean speech data, where multiple frames are associated with each monophone. Monophone alignments are obtained by using ASR after converting the triphone alignments into monophone ones. The counts of each monophone are gathered in a histogram and the most frequent $B_s$ monophones in an utterance are picked up from the dictionary. For each phoneme, each basis is selected randomly from the multiple frames in the dictionary.

In addition, some utterances that include more various phonemes need more bases than the other utterances. Then, it is possible to pick up the bases of frequently appearing monophones utterance-by-utterance by checking the appearance percentage, instead of selecting the fixed top $B_s$ monophones. These two types of initializations are validated in the experimental section.

### 3.4.4 Initialization of spatial correlation matrices

The separation performance can be improved by initializing $\boldsymbol{H}$ from impulse responses [87], but it is difficult to obtain these types of information a priori. The initial $\boldsymbol{H}$ can however be obtained from roughly separated sounds by using binary masking.

#### 3.4.4.1 Source signal enhancement by using binary masking

Binary masking is a source separation technique that masks spectra in the time-frequency domain based on the phase difference $\theta_{kt}(= \arg(X_2/X_1))$. For each source $l$, when a noise comes from another direction than that of the source, the phase difference will be different from that of the source, $\theta^s_{tl}$. Each source can thus be enhanced by masking power spectra in the

time-frequency bins that have different phases from $\theta^s_{tl}$. The mask $W_{ktl}$ can be set as

$$W_{ktl} = \begin{cases} \epsilon & (\min(|\theta_{kt} - \theta^s_{tl}|, 2\pi - |\theta_{kt} - \theta^s_{tl}|) > \theta_c), \\ 1 & (\text{otherwise}), \end{cases} \tag{3.27}$$

where $\epsilon(> 0)$ is a very small constant and $\theta_c$ is a threshold that can be set a priori. If the source direction is unknown, it can be estimated by various algorithms [29, 37].

#### 3.4.4.2 Initialization by using the cross-spectrum method

The cross-spectrum method estimates the spatial correlation matrix at each frame, $\boldsymbol{H}_{ktl}$, as a multiplication of the $l$-th masked data and its Hermitian transpose [88]. After calculating $\boldsymbol{H}_{ktl}$, the initial $\boldsymbol{H}_{kl}$ for MNMF is set as the expectation $E_t$ of $\boldsymbol{H}_{ktl}$ in order for the estimations to be stable as shown in

$$\boldsymbol{H}_{kl} = E_t\left[\boldsymbol{H}_{ktl}\right] = \frac{1}{\sum_t W^2_{ktl}} \sum_t W^2_{ktl} [X_1(t,k), \ldots, X_N(t,k)]^\top [X_1(t,k), \ldots, X_N(t,k)]^*. \tag{3.28}$$

### 3.4.5 Coupled initialization via cluster-indicator latent variables

Cluster-indicator latent variables $\boldsymbol{Z}$ can explicitly relate the spatial information with the spectral information. Fig. 3.11 shows the system components of the proposed method. The combination of our methods described in Sections 3.4.3 and 3.4.4 provides the initial spatial correlation matrix $\boldsymbol{H}$ and the basis matrix $\boldsymbol{T}$. The left part of $\boldsymbol{H}$ is related to the target and its right part is related to the noise. In addition, the first $B_s$ components of $\boldsymbol{T}$ are speech bases and the remaining ones are noise bases. To relate these matrices, the target parts of $\boldsymbol{Z}$ (the elements at the first row and the first to the $B_s$th columns) and noise parts of $\boldsymbol{Z}$ (the elements at the second row and the $(B_s+1)$th to the $B$th columns) should be set larger than the other parts of $\boldsymbol{Z}$. This initialization of $\boldsymbol{Z}$ strongly combines the target/noise spectral information derived from $\boldsymbol{T}$ and the target/noise spatial information derived from $\boldsymbol{H}$ to achieve their separation.

### 3.4.6 Experimental setups

This section validated the effectiveness of our proposed method on the 2ch track of the fourth CHiME challenge (Details are shown in Section 5.4). In the 2ch track, two channels were randomly sampled from the five channels with frontal direction[4]. Thus, microphone positions were different for every utterance. As conventional speech enhancement methods, we employed the challenge baseline beamformer (BeamformIt, denoted as BF) [91], as well as the minimum variance distortionless response (MVDR) beamformer with precise steering vector estimation [92]. The baseline was the conventional MNMF with random initialization of all matrices except $\boldsymbol{H}$,

[4]The total number of microphones was six but one microphone was located at the backend of the tablet.

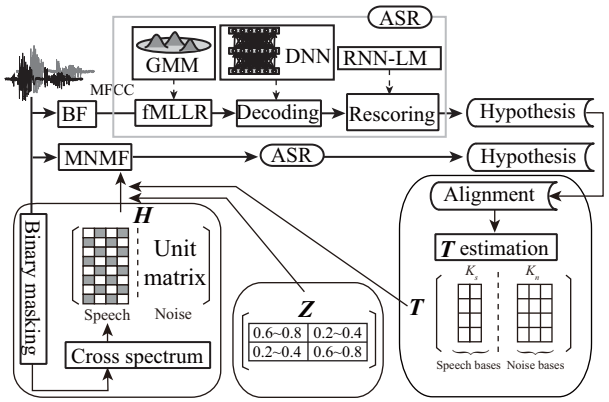



Fig. 3.11 A combination of the ASR system with the proposed MNMF initialization.

Table 3.4 Average WER [%] on the development and test sets of the fourth CHiME challenge for the baseline systems with conventional speech enhancement (SE) methods.

| SE | RNN-LM | Dev | | Test | |
|---|---|---|---|---|---|
| | | real | simu | real | simu |
| None | no | 14.67 | 15.67 | 27.69 | 24.15 |
| | yes | 11.69 | 15.43 | 23.71 | 20.95 |
| BF [91] | no | 10.92 | 12.30 | 20.44 | 19.30 |
| | yes | 8.27 | 9.49 | 16.58 | **15.39** |
| MVDR [92] | no | 10.83 | 11.84 | 19.82 | 19.95 |
| | yes | **7.91** | **9.35** | **15.91** | 16.39 |

which was set to as a unit matrix [86]. There were two outputs of the conventional MNMF and it was necessary to select the appropriate one because it was unknown which one included the target speech. Here, this selection was oracle, i.e., the better hypotheses were selected according to the utterance-based WERs after both were decoded, which is the upper limit performance of the conventional MNMF. Parameter settings of MNMF were as follows: $K = 513$, $B = 30$, and $L = N = 2$, which was common through all the experiments.

For our $\boldsymbol{H}$ initialization, the binary masking assumed that the target speaker was in the frontal position and ideally, the phase differences of the target source $\theta^s$ were near zero, but some errors did occur. For our $\boldsymbol{T}$ initialization, $B_s$ was set to be 20. For the selection involving the top candidates up to a given percentage, the percentage was set such that roughly 20 bases were used on average.

## 3.4.7 Results and discussion

### 3.4.7.1 Baseline and conventional methods

Table 3.4 shows the baseline WERs of the challenge. Baseline BF significantly improved the performance over the unprocessed signals. RNN-LM rescoring reduced the errors by 20%.

Table 3.5 Average WER [%] for the proposed systems; MNMF denotes conventional MNMF where all initial matrices except for $\boldsymbol{H}$ which is set to a unit matrix are random. (I) uses a binary masking based $\boldsymbol{H}$ initialization. (II) is (I) with ASR-based speech bases selection. (III) is (II) with speech bases kept constant during the MNMF update. (IV) is (II) with $\boldsymbol{Z}$ initialization. (V) is (IV) with variable-size speech bases.

| SE | RNN-LM | Dev | | Test | |
|---|---|---|---|---|---|
| | | real | simu | real | simu |
| MNMF | no | 23.99 | 22.42 | 33.98 | 23.18 |
| | yes | 20.96 | 19.62 | 23.46 | 19.49 |
| (I) | no | 10.54 | 10.83 | 18.80 | 15.93 |
| | yes | 7.84 | 8.32 | 14.83 | 12.72 |
| (II) | no | 10.16 | 10.68 | 18.63 | 16.87 |
| | yes | 7.53 | **8.20** | 14.98 | 13.75 |
| (III) | no | 11.00 | 11.16 | 19.65 | 16.03 |
| | yes | 8.08 | 8.68 | 15.91 | 12.42 |
| (IV) | no | 10.00 | 10.77 | 17.88 | 14.48 |
| | yes | 7.42 | 8.26 | 13.97 | 10.99 |
| (V) | no | 9.74 | 10.72 | 17.78 | 14.15 |
| | yes | **7.30** | 8.33 | **13.81** | **10.91** |

MVDR achieved equivalent performance with baseline BF, although in the 6ch track, MVDR outperformed BF [92, 93]. Table 3.5 shows the performance of the conventional MNMF with random initialization, which was even worse than those of the baselines due to spectral distortions introduced by the separation.

#### 3.4.7.2 Proposed methods

Table 3.5 also shows the performance of the proposed method. $\boldsymbol{H}$ initialization ((I) in the table) significantly improved the performance, outperforming both BF and MVDR. Association with $\boldsymbol{T}$ initialization (II) further improved the WER by 0.2–0.4% on the Dev set. Keeping speech bases constant (III) did not improve the performance because there were mismatches between training and test data, thus, updating the bases is necessary. $\boldsymbol{Z}$ initialization (IV) gave additional improvements. Variable-size speech bases (V) improved the performance in some cases but this was not significant. Table 3.6 shows the WER of the respective methods per environment. Our approach was effective for all environments.

Fig. 3.12 shows the standard deviations of the WERs for each speech enhancement. The conventional MNMF had significantly larger standard deviations than the others, which shows the large initial value dependencies. The proposed $\boldsymbol{T}$ initialization (II) decreased the standard deviations and combining $\boldsymbol{Z}$ initialization (IV) achieved the smallest standard deviation.

Table 3.6 WER [%] per environment for each system with RNN-LM rescoring.

| SE | Envir. | Dev | | Test | |
|---|---|---|---|---|---|
| | | real | simu | real | simu |
| None | BUS | 15.25 | 13.55 | 36.19 | 16.40 |
| | CAF | 12.18 | 19.46 | 24.58 | 24.09 |
| | PED | 7.51 | 11.11 | 19.77 | 20.53 |
| | STR | 11.81 | 17.62 | 14.33 | 22.79 |
| BF | BUS | 10.93 | 8.17 | 25.37 | 10.63 |
| | CAF | 8.14 | 12.11 | 15.89 | 18.27 |
| | PED | 5.19 | 7.17 | 13.60 | 15.67 |
| | STR | 8.82 | 10.58 | 11.45 | 16.83 |
| (I) | BUS | 9.59 | 6.92 | 22.81 | 8.20 |
| | CAF | 7.52 | 10.68 | 14.76 | 14.64 |
| | PED | 5.66 | **6.34** | 12.00 | 12.39 |
| | STR | 8.73 | 9.34 | 10.42 | 14.64 |
| (IV) | BUS | 9.78 | 7.29 | 21.95 | 7.71 |
| | CAF | 7.17 | **10.43** | 13.19 | 12.61 |
| | PED | **5.10** | 6.37 | **10.30** | **11.21** |
| | STR | **7.61** | **8.97** | **9.86** | 12.42 |
| (V) | BUS | **8.91** | **6.90** | **21.28** | **7.55** |
| | CAF | **7.02** | 10.96 | **13.02** | **12.01** |
| | PED | 5.35 | 6.50 | 10.91 | 11.23 |
| | STR | 7.90 | 9.16 | 10.03 | **12.14** |

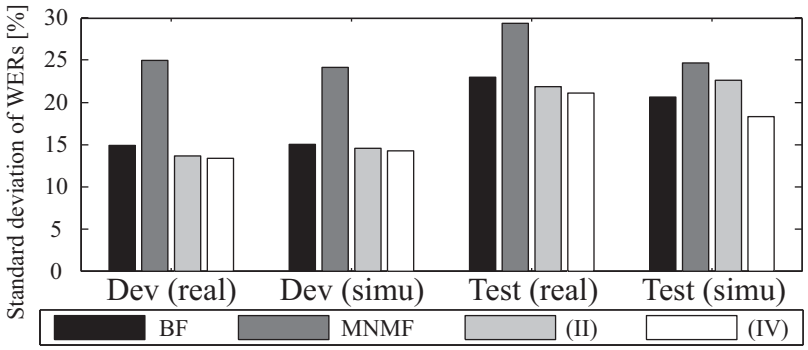



Fig. 3.12 Standard deviations of WERs for each method.

### 3.4.8 Conclusion

This section proposes a combination of binary masking and MNMF. Their spatial correlation matrices were constructed from the sounds roughly separated by binary masking. In addition, it is necessary to coorporate speech enhancement with ASR because the aim of speech enhancement in this case is to improve the ASR performance. The basis matrices corresponding to the speech were initialized from the clean speech based on the ASR results. Third, the cluster-indicator latent variables were initialized to combine the two matrices above. Experimental results on the fourth CHiME challenge show that these initializations were effective for noisy ASR. Compared with the baseline beamformer, although MNMF with random initialization did not improve the WERs, MNMF with the proposed initialization significantly improved the WER.

## 3.5 Voice activity detection (VAD) method based on likelihood ratio test

VAD is an essential pre-process in speech processing [94]. Accurate detection of speech activities effectively reduces error recognition and the adjustment of the strength of noise suppression in noisy environments. The most basic VAD, which assumes that the power of speech is usually greater than that of noise [95], is ineffective in highly noisy environments where speech is masked by noise. The use of the characteristics of speech, e.g., the periodic structure of speech [96], is susceptible to noise [97]. The use of decoder output is effective but computational costs are high [98].

A simple and effective model-based method called the likelihood ratio test (LRT) is effective in highly noisy environments. Even if the power of speech is lower than that of noise, the likelihood of the speech model is greater than that of noise model because the characteristics of speech are available. Sohn *et al.* proposed using the likelihood ratio of speech and noise models after estimating both models from observation to detect speech [47].

Among methods [99, 100, 97, 101, 102] that improve Sohn's method, Fujimoto *et al.* proposed constructing speech models by synthesizing *a priori* clean speech and observed noise at each frame and constructs a noise model by using observed noise to calculate the likelihood ratio of these models [100, 97]. This outperforms Sohn's method, especially in noisy environments, mainly by on-line estimation of models. However, Sohn's method remains an important benchmark of LRT-based VAD and, currently, many comparisons have been made with Sohn's method.

The common point of the above methods is calculating the likelihoods of speech and noise models, respectively, and using the ratio of likelihood to determine whether individual frames are speech or noise. The noise model is estimated from observation and the speech model is estimated by maximum likelihood [47] or by clean speech in advance [97]. In LRT, however, if the likelihood ratio of speech and noise model is estimated directly, the likelihood of individual models is not required.

In the field of machine learning, Sugiyama *et al.* have recently proposed estimating the probability density ratio of two probability distributions directly without estimating their probability densities [103, 104]. This directly models the density ratio function by using a kernel and estimating its parameters from training data, and calculates the likelihood ratio directly, which is effective in change-point detection tasks [105].

There are two problems in applying density ratio estimation to VAD: feature selection and noise adaptation. This is because density ratio estimation puts constraints on feasible features due to the shape of the kernel and speech is dynamic. This section addresses these problems and proposes a method that directly estimates the likelihood ratio for VAD. To use the advantages of conventional LRT and the proposed method, the systems of different features and models are combined. Conventional LRT is introduced in Section 3.5.1 and density ratio estimation in Section 3.5.2. The proposed method is described in Section 3.5.3.

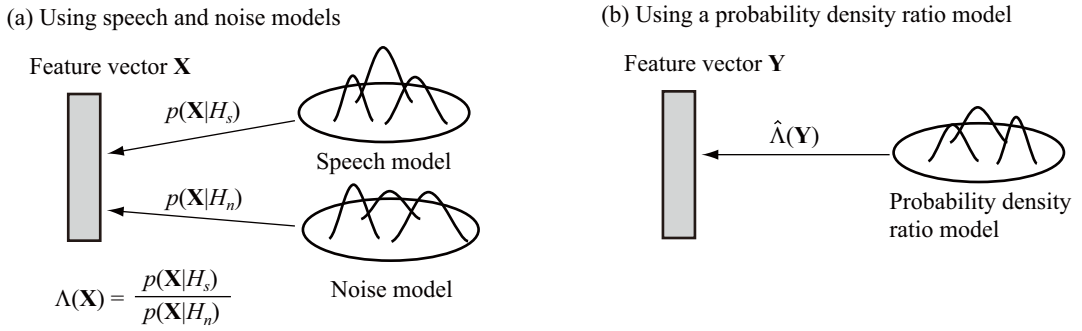



Fig. 3.13 Using speech and noise models and using a probability density ratio model.

### 3.5.1 Likelihood ratio test (LRT)

One of the simplest and most effective conventional likelihood ratio test methods [47] is described here. Let $\boldsymbol{X} = \{X_k\}_{k=1}^{K_X}$ be the observed $K_X$-dimensional spectra. The power spectra $|X_k|^2$ are assumed to be independent conditionally on the noisy speech model $\lambda^S$ in noisy speech frames ($H_S$) and on the noise model $\lambda^N$ in non-speech frames ($H_N$):

$$\begin{aligned} p(\boldsymbol{X}|\lambda^S, H_S) &= \prod_{k=1}^{K_X} \frac{1}{\pi[v_k^S + v_k^N]} e^{-\frac{|X_k|^2}{v_k^S + v_k^N}}, \\ p(\boldsymbol{X}|\lambda^N, H_N) &= \prod_{k=1}^{K_X} \frac{1}{\pi v_k^N} e^{-\frac{|X_k|^2}{v_k^N}}, \end{aligned} \tag{3.29}$$

where $v_k^S$ and $v_k^N$ are the variance of speech and noise spectra, respectively. The log-likelihood ratio of speech and noise at the $k^{\text{th}}$ dimension is then given by

$$\Lambda_k(X_k|\lambda^S, \lambda^N) = \ln \frac{p(X_k|\lambda^S, H_S)}{p(X_k|\lambda^N, H_N)}. \tag{3.30}$$

The geometric mean of the likelihood ratios is used to determine whether individual frames are speech or noise, as

$$\Lambda(\boldsymbol{X}|\lambda^S, \lambda^N) = \frac{1}{K_X} \sum_{k=1}^{K_X} \Lambda_k(X_k|\lambda^S, \lambda^N) \underset{H_N}{\overset{H_S}{\gtrless}} \eta, \tag{3.31}$$

where if $\Lambda(\boldsymbol{X}|\lambda^S, \lambda^N)$ is greater than some threshold $\eta$, the frame is considered to be in a (noisy) speech state, and otherwise in a noise state. The noise model is estimated in advance using observed noise, and the speech model is estimated by maximum likelihood estimation, i.e., $\partial\Lambda_k(X_k)/\partial\lambda_k^S = 0$, which results in the relationship $v_k^S = |X_k|^2 - v_k^N$. This shows that the speech model $\lambda_k^S$ is estimated assuming that the speech and noise powers are additive.

### 3.5.2 Density ratio estimation (KLIEP)

Probability density ratio $q$ for sequential data $y$ is defined as

$$q(y|\lambda^n, \lambda^d) = \frac{p(y|\lambda^n)}{p(y|\lambda^d)}, \tag{3.32}$$

where $p$ is the probability density function of $y$ conditioned on numerator model $\lambda^n$ and denominator model $\lambda^d$, respectively. Here, we assume that training data are labeled as $\boldsymbol{y^n} = \{y^n(i)\}_{i=1}^I$ and $\boldsymbol{y^d} = \{y^d(j)\}_{j=1}^J$ for models $\lambda^n$ and $\lambda^d$, respectively. It is known that simple kernel density estimation, which estimates the density ratio function using statistics of $\boldsymbol{y^n}$ and $\boldsymbol{y^d}$ separately[5], results in low estimation accuracy [105].

The Kullback-Leibler Importance Estimation Procedure (KLIEP) [103], in contrast, directly models density ratio model $\lambda^r$ instead of $\lambda^n$ and $\lambda^d$. This improves the robustness of density ratio calculation. The density ratio is modeled as linear model $\hat{q}(y)$ which consists of $M$ mixture kernels $\varphi_m$, as in Eq. (3.33):

$$\hat{q}(y|\lambda^r) = \frac{\hat{p}(y|\lambda^r, \lambda^d)}{p(y|\lambda^d)} = \sum_{m=1}^{M} \alpha_m \varphi_m(y) = \sum_{m=1}^{M} \alpha_m e^{-\frac{|y-\mu_m^r|^2}{2v^r}}, \tag{3.33}$$

where $\alpha_m$ is a non-negative mixture weight and $\varphi_m$ is a Gaussian kernel whose parameters are $\mu_m^r$ and $v^r$, which are the center and width of a kernel, respectively. A Gaussian kernel requires that the density ratio function takes larger values at the point where many samples from $\boldsymbol{y^n}$ converge, but otherwise takes smaller values close to zero.

Here, $\mu_m^r$, $v^r$, and $\alpha_m$ are unknown variables that are estimated in the following four steps:

(1) Some kernel widths $v^r$ are set arbitrarily.

(2) $M$ samples from $\boldsymbol{y^n}$ are picked as $\{\mu_m^r\}_{m=1}^M$.

(3) Mixture weight $\alpha_m$ is obtained by solving the optimization problem shown below.

(4) The appropriate value of $v^r$ is determined by $n$-fold cross validation.

In KLIEP, $\alpha_m$ is determined as the KL divergence of a sample $y$ from $p(y|\lambda^n)$ to $\hat{p}(y|\lambda^r, \lambda^d)$ is minimized, where $\hat{p}(y|\lambda^r, \lambda^d)$ is the numerator estimated density represented by $\hat{q}(y|\lambda^r)p(y|\lambda^d)$. KL divergence $L$ is represented as

$$L\left(p(y|\lambda^n); \hat{p}(y|\lambda^r, \lambda^d)\right) = \int_{\mathcal{D}} p(y|\lambda^n) \ln \frac{p(y|\lambda^n)}{p(y|\lambda^d)} dy - \int_{\mathcal{D}} p(y|\lambda^n) \ln \hat{q}(y|\lambda^r) dy, \tag{3.34}$$

where $\mathcal{D}$ is a data domain. Since $\hat{p}(y|\lambda^r, \lambda^d)$ is a probability density function, constraint must be satisfied as

$$\int_{\mathcal{D}} \hat{p}(y^n|\lambda^r, \lambda^d) dy^n = \int_{\mathcal{D}} \hat{q}(y^d|\lambda^r) p(y^d|\lambda^d) dy^d = 1. \tag{3.35}$$

[5] For example, after assuming two Gaussian kernels and estimating these parameters from each sample $\boldsymbol{y^n}$ and $\boldsymbol{y^d}$, the ratio of these density functions is calculated [106].

To minimize KL divergence, the second term of Eq. (3.34) is minimized under the constraint in Eq. (3.35) because the first term on the right side of Eq. (3.34) is constant for $\alpha_m$. The optimization problem in Eq. (3.36) is obtained by substituting a sample mean for an expectation of the second terms of Eq. (3.34) and Eq. (3.35). Solving the optimization problem requires only labeled features $\boldsymbol{y^n}$ and $\boldsymbol{y^d}$ and thus does not require information on $\lambda^n$ and $\lambda^d$. This problem is a convex optimization problem because $\alpha_m$ is non-negative and reaches global optimization by gradient descent and constraint satisfaction. Optimized solutions tend to be sparse, that is, some $\alpha_m$ values are zero. This property is effective in reducing computational costs.

$$\begin{aligned} &\underset{\{\alpha_m\}_{m=1}^M}{\arg\min} \left[ -\sum_{i=1}^{I} \ln \left( \sum_{m=1}^{M} \alpha_m \varphi_m(y^n(i)) \right) \right], \\ &\text{subject to } \sum_{m=1}^{M} \alpha_m \left[ \frac{1}{J} \sum_{j=1}^{J} \varphi_m(y^d(j)) \right] = 1. \end{aligned} \tag{3.36}$$

### 3.5.3 Application of KLIEP for VAD

The density ratio estimation is applied to the VAD problem by substituting variables into Eq. (3.33) as

$$y \leftarrow \hat{X}_k, \hat{q} \leftarrow \hat{\Lambda}_k, \tag{3.37}$$

where $\hat{X}_k$ is a component of the $K_{\hat{X}}$-dimensional feature vector $\boldsymbol{Y} = \{\hat{X}_k\}_{k=1}^{K_{\hat{X}}}$ and $\hat{\Lambda}_k$ is a likelihood ratio conditioned on density ratio model $\lambda^r$ obtained by the above procedure[6]. Speech is detected as

$$\hat{\Lambda}(\boldsymbol{Y}|\lambda^r) = \frac{1}{K_{\hat{X}}} \sum_{k=1}^{K_{\hat{X}}} \hat{\Lambda}_k(\hat{X}_k|\lambda^r) \underset{H_N}{\overset{H_S}{\gtrless}} \eta. \tag{3.38}$$

There are two problems in applying the above KLIEP to VAD: the feature selection and the noise adaptation. This is because density ratio estimation puts constraints on feasible features due to the shape of the kernel and speech is dynamic. First, we consider feature selection. Features are assumed to be independent at each dimension. Features are certainly correlated across feature dimensions, but use of a full covariance matrix requires extremely large computational costs. Thus, the density ratio function is estimated at each dimension. Training data need to be labeled as speech and noise. The estimation performance of KLIEP is high when the variance of the denominator distribution $v^d$ is greater than that of the numerator distribution $v^n$, because the value of the density ratio function is unstable when the denominator value is small and the numerator value is large. If, for example, denominator and numerator distributions are represented as Gaussians kernels ($\exp(-|y-\mu^d|^2/2v^d)$ and $\exp(-|y-\mu^n|^2/2v^n)$), the density ratio function is represented as $\exp(-|y-\mu^r|^2/2v^r)$ where

$$\mu^r = \frac{v^d\mu^n - v^n\mu^d}{v^d - v^n},\ v^r = \frac{v^n v^d}{v^d - v^n}.$$

[6]We refer to a Matlab® code [107] when implementing model learning.

In this case, estimation is only stable when $v^d$ is greater than $v^n$. Otherwise, $v^r$ is negative. Power often satisfies this requirement because the dynamics of noise is greater than that of speech in the long term whereas the MFCC feature, which is normally used for ASR, does not necessarily satisfy this requirement. In fact, in the case of MFCC, the estimation accuracy of the density ratio function is low as shown in Section 3.5.7.1. We propose to use a log power spectrum as $\hat{X}_k = \ln |X_k|^2$ for the feature because the range of a 'raw' power spectrum is too large to be represented by a linear model.

Second, we consider the adaptation of a model. There is a mismatch between training and evaluation environments due to noise diversity. Adaptation of a model is effective because speech and noise are dynamic [108]. For both Sohn's method and the proposed method, it is necessary to adjust the mean of features because these methods assume a relative power difference between speech and noise. It is clearly shown that, even for the same speech, the boundary of speech and noise shifts if microphone gain changes. Sohn's method avoids this mean shift effect by using variance as a model. The proposed method equates the mean and variance of noise during first $N_N$ frames with those of training noise to adapt noise. The on-line adaptation of a model, e.g., [97, 109], is a future work.

### 3.5.4 Combination of VAD systems

As [97] mentioned, combining different features and models is effective. Here, the proposed method is combined with Sohn's method to exploit the advantages of both. Two likelihood ratios are combined as

$$\Lambda'' = \gamma \Lambda'(\boldsymbol{X}|\lambda^S, \lambda^N) + (1-\gamma)\hat{\Lambda}'(\hat{\boldsymbol{X}}|\lambda^r), \tag{3.39}$$

where $\Lambda'$ and $\hat{\Lambda}'$ are likelihood ratios normalized by the maximum value of $\Lambda$ and $\hat{\Lambda}$ during utterance and $\gamma$ is the constant weight of the two systems, which weighs the importance on either system ($\gamma = 0$: Sohn's method and $\gamma = 1$: proposed method). VAD is performed using the obtained likelihood ratio $\Lambda''$

### 3.5.5 Automatic thresholds determination

Thresholds $\eta$ must be set appropriately for VAD. There seems to be no studies on automatic thresholds determination. The optimized thresholds are dependent on environments and are difficult to set. We propose a method that automatically sets thresholds using clustering analysis. An initial value $\eta_0$ is set for the first state because thresholds cannot be calculated with just noise information. After speech detection, clustering of previous likelihood ratios (e.g. $\ln \Lambda$ for Sohn's method) is done. For example, they are divided into $N_{cl} (\geq 3)$ clusters using the $K$-means algorithm, as shown in Fig. 3.14. In this case, speech data is leaned against the larger value of $\ln \Lambda$ and noise data is leaned against the smaller value. The thresholds $\eta$ can then calculated by taking a middle value of the speech and noise clusters. For example, the threshold can be

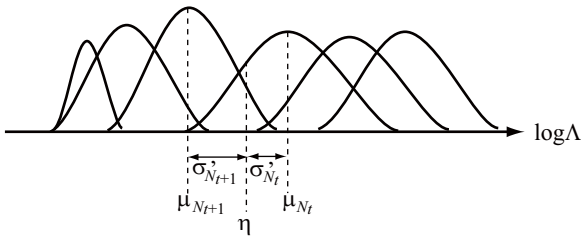



Fig. 3.14 Determination of the threshold $\eta$ using $N_{cl}$ clusters.

determined by calculating each mean $\mu_j$ and variance $v_j^{r\prime}$ $(1 \leq j \leq N_{cl})$ of the clusters, sorting them by $\mu_j$, and taking an inner point between the $N_t^{\text{th}}$ and $(N_t+1)^{\text{th}}$ clusters.

### 3.5.6 Experimental setups

The proposed method was evaluated using the CENSREC-1-C database [110, 68], which is commonly used for evaluating VAD in noisy environments. Evaluation data were recorded in two real environments: 'RESTAURANT' (speech and foot noise: non-stationary) and 'STREET' (traffic noise: stationary), with two different SNRs ('HIGH' and 'LOW'). There were ten speakers (five males and five females). There were four files containing 38-39 utterances for each speaker. Each file consisted of 8-10 utterances, which were 1-12 digit numbers. The sampling frequency was 8 kHz and the dimension, the window length and the frame shift of short-time Fourier transform were 256, 25 ms and 10 ms, respectively. Feature dimensions, $K_X$ and $K_{\hat{X}}$ were 129, considering symmetry. Performance was evaluated in terms of the correct and accuracy rate [%].

We compared results for the proposed method to those of two conventional methods: a power-based method attached to CENSREC-1-C as a baseline (similar to [95]) and Sohn's method in Section 3.5.1. Some methods that use on-line adaptation for noise certainly outperform Sohn's method because, for this database, noise adaptation is effective due to the long files which contain multiple utterances with changing noise. However, Sohn's method is still an LRT based benchmark among methods without on-line adaptation and the proposed method does not use on-line adaptation.

The first 10 $(= N_N)$ frames were used to construct a noise model for Sohn's method and to adapt the mean and the variance of a background noise for the proposed method. Thresholds $\eta$ were optimized among some candidates. The density ratio model was trained using CENSREC-4 database [68], including eight types of reverberation and noise with SNRs {5, 10, 20, 25, 30} [dB], which were totally different from CENSREC-1-C. They were down-sampled to 8 kHz from 16 kHz. The number of training data was 16000 (160 seconds) for speech and noise data, respectively. The number of kernels was 20 $(= M)$. The width $v^r$ was determined by 5-fold cross-validation. The weight $\gamma$ for system combination was 0.3 (turned on the preliminary experiments).

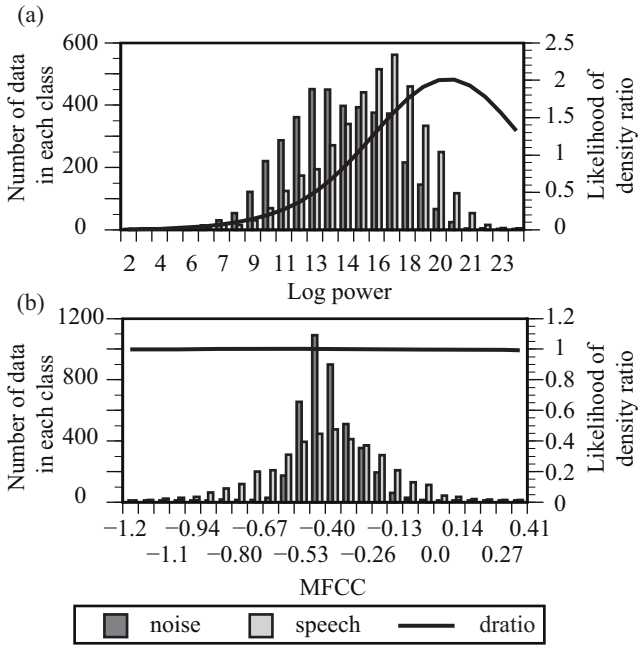



Fig. 3.15 Histogram of (a) the log power ($15^{\text{th}}$ dimension) and (b) MFCC ($1^{\text{st}}$ dimension) of speech and noise and probability density ratio (dratio).

## 3.5.7 Results and discussion

### 3.5.7.1 Density ratio estimation

Fig. 3.15 (a) shows the distributions of the $15^{\text{th}}$ dimension (which approximately equals to 500 Hz and includes rich information of speech) of the log power of speech and noise and a density ratio function obtained by KLIEP, where there are 13 non-zero $\alpha_m$. This shows that KLIEP estimated a density ratio function for VAD. On the other hand, Fig. 3.15 (b) shows the distributions and a density ratio function of the $1^{\text{st}}$ dimension of MFCC. Here, because $v^d$ is apparently much smaller than $v^n$, the density ratio function does not satisfy KLIEP requirements. The estimated function shape is flat and cannot discriminate between speech and noise.

Table 3.7 shows that the proposed method improves the average correct rate by 28.6% from the CENSREC baseline and 6.0% from Sohn's method, and improves the average accuracy rate by 74.8% and 8.5%, respectively. The proposed method outperforms the conventional methods for 'RESTAURANT', which is non-stationary noise. This shows that the density ratio model is more robust in mis-estimating the model than Sohn's model. The system combination, moreover, improves the accuracy rate by 13.9% from Sohn's method. Sohn's method is effective in stationary noise, therefore the system combination exploits the advantages of both Sohn's method and the proposed method.

Fig. 3.16 (a) and (b) show the likelihood ratio calculated by Sohn's method and the proposed method, respectively, under the condition of RESTAURANT(HIGH). The likelihood ratio of

Table 3.7 Correct and accuracy rates[%] of the proposed method (prop) and system combination (comb) compared to those of the CENSREC-1-C baseline (base) and Sohn's method (Sohn) in terms of environments (RESTAURANT and STREET) and SNR (high (H) and low (L)).

| | | Correct | | | | Accuracy | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | base | Sohn | prop | comb | base | Sohn | prop | comb |
| RESTRANT | H | 74.2 | 73.0 | **89.0** | 81.2 | 21.5 | 41.5 | **67.0** | 57.1 |
| | L | 56.5 | 59.4 | **63.5** | 57.4 | -43.5 | 13.9 | 15.9 | **24.6** |
| STREET | H | 39.4 | 94.2 | 91.0 | **95.7** | -15.7 | 86.1 | 82.6 | **92.5** |
| | L | 41.5 | 75.4 | 82.6 | **86.1** | -33.9 | 52.2 | 62.0 | **74.8** |
| Average | | 52.9 | 75.5 | **81.5** | 80.1 | -17.9 | 48.4 | 56.9 | **62.3** |

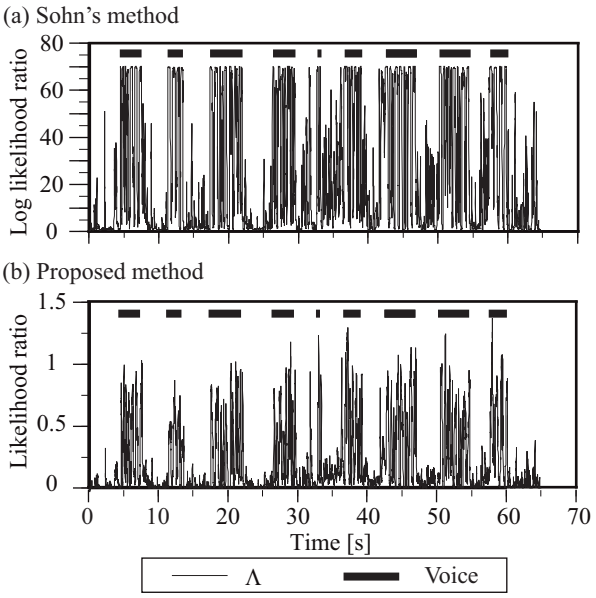



Fig. 3.16 Likelihood ratio of (a) Sohn's method and (b) the proposed method (RESTAURANT(HIGH)).

the proposed method remains stable at low during the non-speech area. Because noise is non-stationary, Sohn's noise model obtained by using the first 10 frames mismatches actual noise and generates high likelihood ratios that lead to a false detection. The proposed method is more robust than Sohn's method for mis-estimation by using a density ratio model.

#### 3.5.7.2 Automatic thresholds determination

To validate the performance of automatic determination of thresholds, Figs. 3.16(a) and (b) show the relationship between the likelihood ratio calculated by 'Sohn' and thresholds calculated by the proposed method in 'RESTAURANT (HIGH)' and 'STREET (LOW)', respectively. The

initial value $\eta_0$ was set to 60. The threshold stays high for non-stationary noise (a) to reduce the misdetection and low for stationary noise (b) to reduce the misrejection.

### 3.5.8 Conclusion

We proposed a voice activity detection method based on the density ratio estimation. Experiments show that the proposed method is more effective than conventional methods, especially under non-stationary noisy environments.

## 3.6 ASR performance estimation of clipped speech

Recently, speech recognition systems have been used in various environments. As a result, we are facing low-quality speech data. One of the factors that degrade speech quality is clipping mainly caused by inappropriate microphone settings. Apparently, clipping degrades speech quality, and a study [111] has shown that speech recognition performance deteriorates for real data. However, in that study [111], other factors, such as contents of utterances, recording devices or speakers, are uncontrolled and the influence of clipping itself on speech recognition performance cannot be evaluated quantitatively. In this study, we clarify the relationship between clipping level, which represents the extent of clipping, and automatic speech recognition (ASR) performance for various artificially clipped utterances by changing the clipping level but keeping the other factors the same.

Practically, ASR performance estimation is useful. There are some studies on estimating the ASR performance in reverberant and noisy environments [112, 113] but, to the best of our knowledge, there are as yet no studies on estimating the ASR performance of clipped speech. Our experiments show that the ASR performance can be expressed as a logistic regression using the SNR and perceptual evaluation of speech quality (PESQ) [114]. Moreover, we show the explicit relationship between SNR and clipping level, theoretically.

### 3.6.1 Clipped signals and clipping level

Clipped signals $\hat{y}$ are obtained by clipping original signals $y$ with a clipping level $\theta_c$, as shown in Eq. (3.40), after $y$ is normalized between $-1$ and 1.

$$\hat{y} = \begin{cases} \mathrm{sign}(y)\theta_c & (|y| \geq \theta_c) \\ y & (\text{otherwise}) \end{cases} \tag{3.40}$$

Here, "sign" returns to 1 if the argument is positive or to $-1$ otherwise. When $\theta_c$ is one, $\hat{y}$ equals $y$.

### 3.6.2 Signal-to-noise ratio (SNR) estimation of clipped speech

For clipped speech, SNR $\psi$ is a function of $\theta_c$,

$$\psi(\theta_c) = 10 \log_{10} \frac{\sum_{t \in \mathcal{T}_c} y_t^2}{\sum_{t \in \mathcal{T}_c} (y_t - \hat{y}_t)^2}, \tag{3.41}$$

where $t$ is the sample number and $\mathcal{T}_c$ is a set of the indexes of clipped samples.

In accordance with Ref. [115], we assume that a background model of speech is represented as a Laplacian,

$$p(y) = \frac{1}{2b} \exp\left(-\frac{|y|}{b}\right), \tag{3.42}$$

where $2b^2$ is the variance of the Laplacian. The expectation of the numerator of Eq. (3.41) is given as

$$\begin{aligned}\sum_{t\in\mathcal{T}_c} y_t^2/N_c &= \int_{\theta_c}^{1} y^2 p(y)dy + \int_{-1}^{-\theta_c} y^2 p(y)dy, \\ &= 2\int_{\theta_c}^{1} y^2 p(y)dy = ((\theta_c+b)^2+b^2)e^{-\frac{\theta_c}{b}} - (b_1^2+b^2)e^{-\frac{1}{b}},\end{aligned} \tag{3.43}$$

where $N_c$ is the number of clipped samples and $b_1$ is $1+b$. Here, we use the relations below:

$$\begin{aligned}f_1 &= \int_{\theta_c}^{1} p(y)dy = \frac{1}{2}\left(e^{-\frac{\theta_c}{b}} - e^{-\frac{1}{b}}\right), \\ f_2 &= \int_{\theta_c}^{1} yp(y)dy = \frac{1}{2}\left(\theta_c e^{-\frac{\theta_c}{b}} - e^{-\frac{1}{b}}\right) + bf_1, \\ f_3 &= \int_{\theta_c}^{1} y^2 p(y)dy = \frac{1}{2}\left(\theta_c^2 e^{-\frac{\theta_c}{b}} - e^{-\frac{1}{b}}\right) + 2bf_2.\end{aligned} \tag{3.44}$$

On the other hand, the expectation of the denominator is given as

$$\begin{aligned}\sum_{t\in\mathcal{T}_c} \frac{(y_t-\hat{y}_t)^2}{N_c} &= \int_{\theta_c}^{1} (y-\theta_c)^2 p(y)dy + \int_{-1}^{-\theta_c} (y-(-\theta_c))^2 p(y)dy, \\ &= 2\int_{\theta_c}^{1} (y-\theta_c)^2 p(y)dy = 2b^2 e^{-\frac{\theta_c}{b}} - (b_1^2+b^2-\theta_c(2b_1-\theta_c))e^{-\frac{1}{b}}.\end{aligned} \tag{3.45}$$

Substituting Eqs. (3.43) and (3.45) with Eq. (3.42) into Eq. (3.41), SNR $\psi$ becomes a function of two parameters, $b$ and $\theta_c$, as

$$\psi_b(\theta_c) = 10\log_{10}\frac{\left((\theta_c+b)^2+b^2\right)e^{\frac{1-\theta_c}{b}} - b_1^2 - b^2}{2b^2 e^{\frac{1-\theta_c}{b}} - b_1^2 - b^2 + \theta_c(2b_1-\theta_c)}. \tag{3.46}$$

Note that $b$ can be derived by the shape fitting of nonclipped speech *a priori*.

### 3.6.3 ASR performance estimation by logistic regression

The ASR performance (word recognition accuracy) $A$ can be estimated, using a logistic regression [112], as

$$A(x) = \frac{\alpha}{1+\exp(-\beta(x-\gamma))}, \tag{3.47}$$

where $\alpha$, $\beta$, and $\gamma$ are parameters of regression and $x$ is an objective measure. In this section, $x$ is the SNR in Eq. (3.41) and PESQ, which evaluates speech quality, ranges from 0.5 to 4.5 [114]. Under noisy environments, a previous study [112] revealed that PESQ has a good correlation with the ASR performance. $\alpha$ can be determined from the accuracy, $A_o$, for original (nonclipped) speech as

$$\alpha = \begin{cases} A_o & \text{(for SNR)}, \\ A_o\left(1+\exp(-\beta(4.5-\gamma))\right) & \text{(for PESQ)}, \end{cases} \tag{3.48}$$

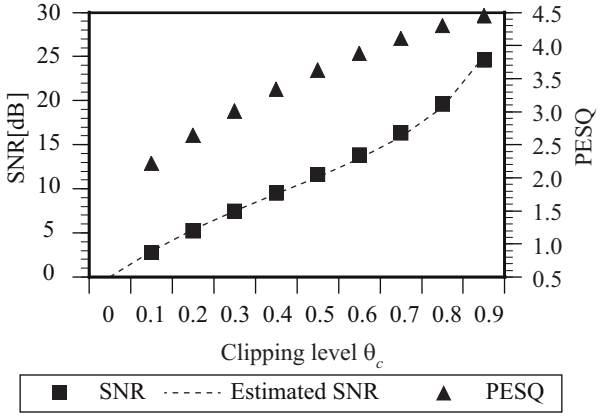



Fig. 3.17 SNR and PESQ of clipped signals in terms of clipping level $\theta_c$, which is defined in Eq. (3.40), with SNR estimated using Eq. (3.46).

because $A(\infty)$ for SNR and $A(4.5)$ for PESQ must be equal to $A_o$. $\beta$ and $\gamma$ are estimated by the minimum square errors criterion. $\alpha$ represents the difficulty of the task; $\beta$ is the sensitivity of the performance against $x$; $\gamma$ is the point where recognition accuracy becomes half, i.e., the point of inflection, and indicates the robustness against clipping.

### 3.6.4 Experimental setups

We evaluated the JEIDA-JCSD (B-set) dataset consisting of 100 Japanese city names (e.g., "Sapporo") spoken by 20 male and 20 female speakers. Sampling frequency was 16 kHz. Clipped utterances were artificially made with different $\theta_c$ values in the range from 0.1 to 0.9. We prepared two tasks, a large vocabulary (155,592 words) task and a small vocabulary (100 words) task, because the influence of clippings can depend on the difficulty of the task. We used the Julius (ver. 4.2.1) software [116] for decoding. The acoustic model was the 64 mixture phonetic-tied mixture [117] context-dependent triphone HMM attached to Julius. The number of states was 3,131, and the number of Gaussians was 8,256. The acoustic features were 12-dim MFCCs, their $\Delta$, and $\Delta$ log power. The total number of dimensions was 25.

### 3.6.5 Results and discussion

#### 3.6.5.1 ASR performance estimation

Fig. 3.17 shows the SNR and PESQ for clipped utterances. SNR and PESQ had almost linear relationships with clipping level, and they dropped with decreased clipping level. In the figure, the dotted line shows the SNR estimated using Eq. (3.46) with $b = 0.15$. The estimated SNR matched the actual SNR.

Table 3.8 Parameters used in performance estimation.

| | SNR | | PESQ | |
|---|---|---|---|---|
| | Large | Small | Large | Small |
| $A_o$ | 63.0 | 93.0 | 63.0 | 93.0 |
| $\beta$ | 0.3 | | 1.7 | |
| $\gamma$ | 0 | 5.8 | 1.8 | 2.8 |

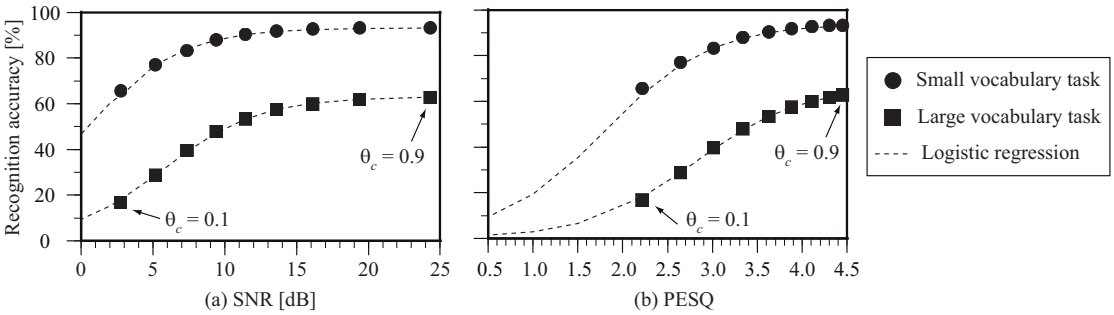



Fig. 3.18 Recognition rate for small vocabulary (100 words) and large vocabulary (155,592 words) tasks in terms of (a) SNR and (b) PESQ. Clipping level $\theta_c$ varies from 0.1 to 0.9. For speech recognition performance estimation, logistic regression was used.

Fig. 3.18 shows the relationships of (a) SNR and (b) PESQ with word recognition accuracy. Clipping levels $\theta_c$ were arranged from 0.1 to 0.9. When $\theta_c$ was over 0.7, the recognition accuracy did not decrease significantly and speech quality degradation was inaudible. Otherwise, the recognition accuracy decreased; this tendency was more significant for the large vocabulary task. SNR and PESQ had clear correlations with the word recognition accuracy. The dotted line in the figures is a fitted logistic regression curve estimated by the minimum square errors criterion. Table 3.8 shows the parameters. A logistic regression can predict word recognition accuracies well with a small number of parameters.

#### 3.6.5.2 Phonemes error tendencies

Fig. 3.19 shows the confusion matrices of 23 Japanese phonemes (a) without clippings and (b) with clippings, respectively. The number of errors increased among vowels. The number of errors where consonants were recognized as vowels was greater than that of errors where vowels were recognized as consonants.

The number of insertion errors increased more than that of deletion errors, especially for consonants. The whitening effect due to clipping caused the voices to become similar to the noise, and the voices did not match the acoustic models. Consonants were more susceptible to these phenomena than vowels, because consonants are originally more similar to the noise than vowels.

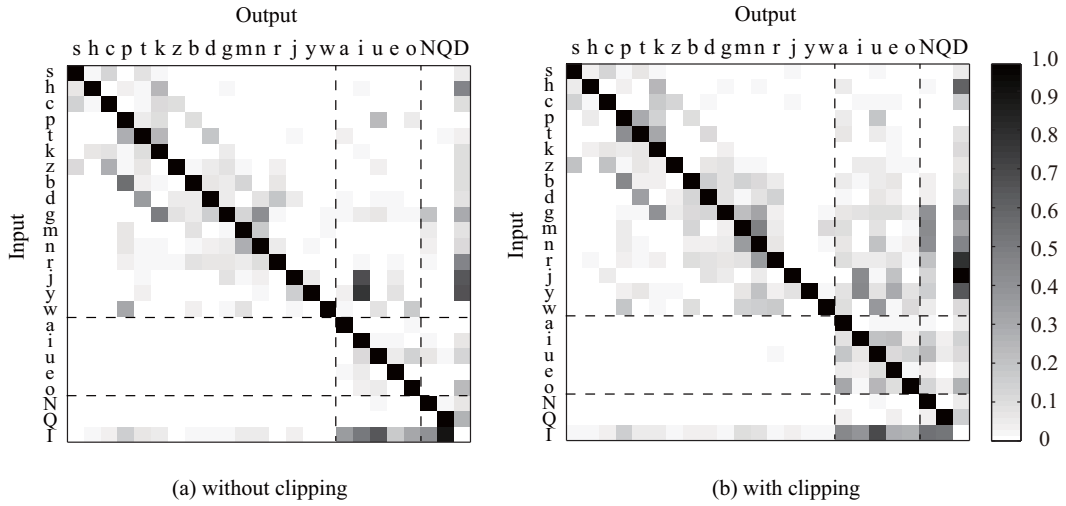



Fig. 3.19 Confusion matrices of Japanese phonemes (a) without clippings and (b) with clippings ($\theta_c = 0.2$), where 'I' stands for insertion and 'D' stands for deletion. Contours show the ratio of respective elements. The sum along each line is normalized to one.

### 3.6.6 Conclusion

Speech recognition performance was evaluated by changing the clipping level while keeping the other factors the same. Experiments showed that SNR and PESQ correlate with the ASR performance of clipped speech as well as noisy speech cases. Moreover, we derived a theoretical estimation formula of SNR after clipping and showed its high accuracy. Confusion matrices revealed that some typical errors were caused by clipping, and consonants were more susceptible to clipping than vowels.

## 3.7 Compensation of mismatched sampling frequency

Broad-band speech improves the performance of ASR, but the performance is significantly degraded when broad-band speech is used for training acoustic models and narrow-band speech is input for ASR decoding. Many bandwidth extension (BWE) methods have been proposed for improving a perceptual subjective impression. One of the most effective BWE methods is a GMM-based BWE [118]. On the other hand, recently, neural network-based signal restoration methods have been widely used. Recurrent structures are effective for speech enhancement and, in particular, a long short-term memory recurrent neural network (LSTM-RNN) [119] has high reconstruction performance for signal restoration. In this letter, we propose to use the LSTM-RNN for BWE, and its performance is evaluated for the TIMIT phoneme recognition task.

using NMF [120]

### 3.7.1 Gaussian-mixture-model-based bandwidth extension (BWE)

In the field of voice conversion, where the speech of one speaker is converted into that of another speaker, a GMM-based voice conversion technique has been proposed [121]. This type of GMM-based voice conversion is applied to a BWE task [118]. We use this method as a baseline. In the context of voice conversion, a narrow-band speech is the original speech and a broad-band speech is the converted speech. Full covariance GMMs are used for modeling concatenated feature vectors before and after BWE, as shown in Fig. 3.20. Converted, i.e., BWE, speech is estimated on the basis of the maximum likelihood criteria,

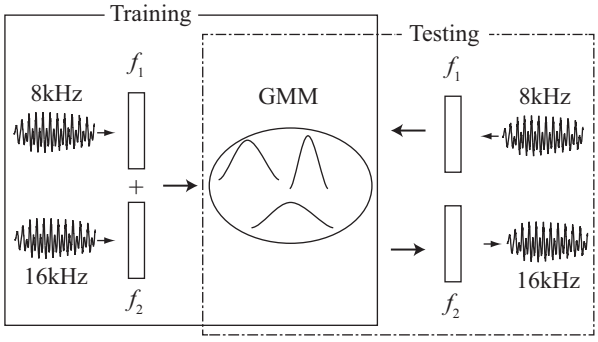



Fig. 3.20 GMM-based BWE.

### 3.7.2 Long short-term memory recurrent-neural-network(RNN)-based BWE

Generally, for time-series signals, RNNs have higher performance than simple NNs because recurrent structures can consider time-series information. The LSTM-RNN has been proposed

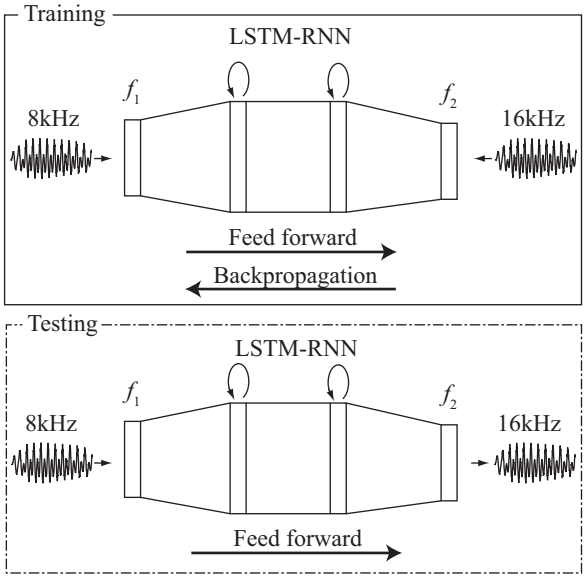



Fig. 3.21 LSTM-RNN-based BWE.

to relax the influence of vanishing gradient problems in the RNN and to deal with longer contexts [119]. Its effectiveness has been shown for a speech enhancement task [122]. The LSTM-RNN with a narrow-band speech input and broad-band speech output is trained by an error backpropagation method based on a least-square-error criterion, as shown in Fig. 3.21.

### 3.7.3 Experimental setups

For BWE, two types of speech features were extracted: 1) mel cepstrum (mcep), which is widely used for speech synthesis [123], and 2) MFCC, which is widely used for ASR. In the case of a GMM-based BWE, 1) the dimensions of mcep features were 17 for 8 kHz and 25 for 16 kHz, and a total of 84-dimensional features in conjunction with their $\Delta$ features were used. For ASR, MFCC features were extracted after signal waves in the time domain were restored from mcep features. 2) The dimension of MFCC features was 13 for both 8 and 16 kHz, and a total of 52-dimensional vectors were used with their $\Delta$ features. In this case, the obtained MFCC features were directly input for ASR. SPTK toolkit (ver.3.7)[7] was used.

In the case of an LSTM-RNN-based BWE, 1) the LSTM-RNN was trained to predict 25-dimensional mcep static features with 25-dimensional $\Delta$, i.e., 50-dimensional in total, features for 16 kHz from the 17-dimensional static mcep with 17-dimensional $\Delta$, i.e., 34-dimensional in total, mcep features for 8 kHz. 2) For MFCC, 26-dimensional MFCC features comprising 13-dimensional static features and $\Delta$ features, were used. The "currennt" toolkit (ver.0.2)[8] [122]

[7] http://sp-tk.sourceforge.net/
[8] https://sourceforge.net/projects/currennt/

Table 3.9 Phoneme error rate (PER) [%] on the **dev** set of the TIMIT phoneme recognition task, evaluating 16 kHz and 8 kHz speech. 8 kHz speech was recognized using 16 kHz and 8 kHz models. MFCC features were used for ASR with advanced ASR techniques such as feature transformation (LDA+MLLT) and speaker adaptation (fMLLR).

| | | ASR feature | | |
|---|---|---|---|---|
| eval | train | MFCC | +LDA+MLLT | +fMLLR |
| 16k | 16k | 23.1 | 21.3 | 18.9 |
| 8k | 16k | 32.3 | 28.8 | 23.4 |
| 8k | 8k | 23.5 | 22.5 | 20.3 |

was used.

The training data of the BWE model and ASR acoustic model were the same as the training data of the TIMIT phoneme recognition task, which was one of the most standard corpora for English ASR. Their performances were evaluated using the development set and evaluation set of the TIMIT in terms of phoneme error rate (PER) using the Kaldi toolkit[9] [124]. For ASR, maximum-likelihood GMM acoustic models were used with MFCC+$\Delta$+$\Delta^2$ features. To improve the ASR performance, two types of advanced ASR techniques were used. The first one was feature transformation by linear discriminant analysis (LDA) [125] and maximum-likelihood linear transformation (MLLT) [126]; the second one was speaker adaptation as shown in Section 4.2.3.2.

### 3.7.4 Results and discussion

Table 3.9 shows the baseline performance on the development set. The performance was the highest for the matched case of recognizing 16 kHz speech with 16 kHz models. The second best one was the matched case of recognizing 8 kHz speech with 8 kHz models. These matched cases showed much better performance than the mismatched case of recognizing 8 kHz speech with 16 kHz models. Speaker adaptation decreased the performance gaps between matched and mismatched conditions. When sampling frequencies are different between the training and the test speech, speaker adaptation compensates for the influence of mismatch to some extent but the recognition performance was significantly worse than those of matched conditions.

Table 3.10 shows the performance after BWE for mcep features. Both gender-dependent and gender-independent BWE models were prepared, but their performance differences were small for both the GMM and LSTM-RNN cases. For all cases, the LSTM-RNN outperformed the GMM. This shows the effectiveness of a LSTM-RNN-based BWE, as in the case of speech enhancement.

Table 3.11 shows the performance of directly predicted MFCC features. Gender-independent models were used in the experiments below. There are two cases: without and with mean normalization of input features to the GMM or LSTM-RNN. Mean normalization was essential for the GMM and effective for LSTM. In the two cases of the GMM without speaker adaptation, the performance was degraded, but in the other cases, direct estimation of MFCC improved the

[9] http://kaldi.sourceforge.net/

Table 3.10 PER [%] on the **dev** set, evaluating GMM- and LSTM-RNN-based BWE (8 kHz→ 16 kHz). Mel-cepstrum features were used for BWE. Both gender-dependent (gd) and gender-independent (gi) models were constructed.

| | ASR feature | | | | | |
|---|---|---|---|---|---|---|
| | MFCC | | +LDA+MLLT | | +fMLLR | |
| gd/gi | gd | gi | gd | gi | gd | gi |
| GMM | 28.9 | 29.3 | 27.7 | 27.9 | 25.2 | 25.4 |
| LSTM | 25.5 | 25.5 | 23.8 | 23.9 | 22.0 | 22.1 |

Table 3.11 PER [%] on the **dev** set. MFCC features without and with mean normalization (Mean norm.) were used for BWE.

| | ASR feature | | | | | |
|---|---|---|---|---|---|---|
| | MFCC | | +LDA+MLLT | | +fMLLR | |
| Mean norm. | - | ✓ | - | ✓ | - | ✓ |
| GMM (gi) | 36.0 | 30.4 | 35.3 | 29.0 | 31.9 | 24.7 |
| LSTM (gi) | 25.5 | 24.7 | 24.1 | 23.0 | 21.5 | 20.7 |

Table 3.12 PER [%] on the **test** set. Mel-cepstrum features (mcep) and MFCC features were used for BWE.

| | | ASR feature | | | | | |
|---|---|---|---|---|---|---|---|
| | | MFCC | | +LDA+MLLT | | +fMLLR | |
| eval | train | Baseline | | | | | |
| 16k | 16k | 24.9 | | 22.3 | | 19.9 | |
| 8k | 16k | 34.8 | | 30.4 | | 25.3 | |
| 8k | 8k | 25.1 | | 23.5 | | 21.0 | |
| | | BWE | | | | | |
| BWE feature | | mcep | MFCC | mcep | MFCC | mcep | MFCC |
| GMM (gi) | | 31.4 | 32.6 | 29.8 | 29.9 | 26.6 | 26.1 |
| LSTM (gi) | | 27.2 | 25.9 | 25.2 | 24.0 | 23.4 | 21.9 |

performance compared with that of the mcep-based BWE. For the purpose of ASR, a direct estimation of the features suitable for ASR was effective. BWE improved the performance of ASR for 8 kHz speech without switching acoustic models.

Table 3.12 shows the results of the test set. The tendencies were similar to those of the development set. The LSTM-RNN outperformed the GMM. LSTM using MFCC features achieved the best performance, where the differences between matched cases and BWE cases were less than 1%.

There is an advantage of the proposed method compared with the use of the matched 8 kHz acoustic model. The proposed method does not require an acoustic model change; thus, it can be widely used for various existing ASR systems without troublesome acoustic model training. If matched acoustic models are needed, training for both 16 kHz and 8 kHz is needed. The training time of acoustic models doubles for each ASR system, whereas the training of the proposed

BWE model is required only once. Constructing two types of acoustic models for each system is inefficient because 16 kHz speech has recently come more frequent than 8 kHz speech.

### 3.7.5 Conclusion

We proposed the LSTM-RNN-based BWE and compared its performance with that of a conventional GMM-based BWE in an ASR experiment. Experiments using the TIMIT corpus showed that LSTM-RNN-based BWE was more effective than GMM-based BWE and that predicting MFCC features directly was better than predicting mel-cepstrum features for ASR purposes. The LSTM-RNN achieved a performance equivalent to those of matched cases without the need to switch acoustic models.

## 3.8 Conclusion of the chapter

This chapter deals with front-end techniques. For single-channel case, SS-based dereveberation methood was proposed in Section 3.2. For multi-channel case, combination of BM and IVA was proposed in Section 3.3 and effective initialization method for MNMF was proposed in Section 3.4. In addition, VAD was improved by using density ratio estimation. Finally, the influences of clipping and mismatched sampling frequencies were experimentally investigated in Sections 3.6 and 3.7.

Journal papers related to this chapter are [127, 128, 129, 27] and conference papers are [130, 131].

# 4 Back-end techniques for robust ASR

## 4.1 Introduction

Recent improvements in back-end techniques render ASR systems [132, 133] more accurate. Robust back-end methods are as important as robust front-end ones. Fig. 4.1 shows back-end components. After acoustic features are extracted, feature transformation transforms it to features which are easier to deal with. Various types of feature transformations have been proposed [125, 126, 134, 135, 136, 137]. Widely used feature transformation approaches are linear discriminant analysis (LDA) [125] and maximum likelihood linear transformation (MLLT) [138, 139, 126, 134]. LDA makes use of long context features across a few contiguous frames (e.g., nine frames) to exploit feature dynamics, which reduces the influence of non-stationary noises and reverberation. MLLT finds a linear transformation to reduce state-conditional feature correlations; it performs a joint optimization of feature transformation matrices and acoustic model parameters. Feature adaptation such as speaker adaptive training (SAT) [135] and feature-space maximum likelihood linear regression (fMLLR) [139] is an effective option. This originally aims to decrease the variation between speakers, but they are also known to improve the ASR accuracy in noisy environments by adapting to unknown and changing noise conditions in effect, performing noise adaptive training [140, 139, 141].

ASR systems are expected to output correct hypotheses. Discriminative training [142, 143] is a framework to correct errors in training. Over the past 20 years in particular, model training techniques have gradually migrated from maximum-likelihood (ML) estimation approaches to discriminative training techniques [144, 145, 146, 147]. This section mainly focuses on discriminative training applied to LDA in Section 4.3.2, acoustic models in Sections 4.4.2 and 4.5.2, and

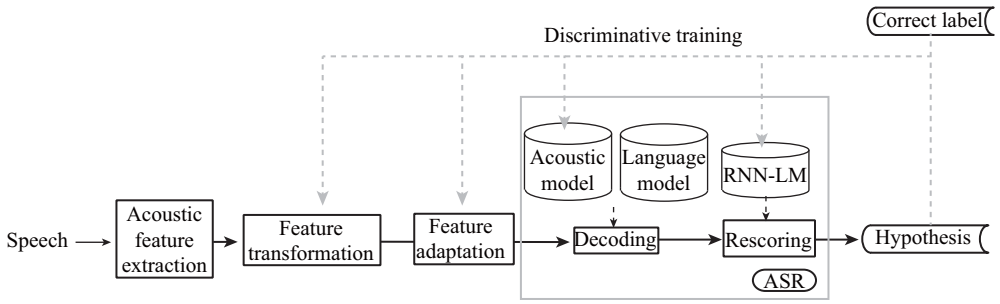



Fig. 4.1 Back-end process appeared in Chapter 4.

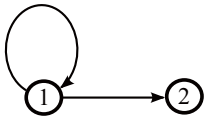



Fig. 4.2 An example of HMM.

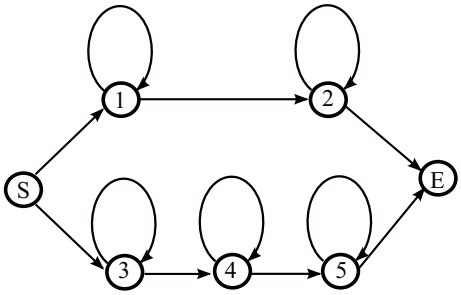



Fig. 4.3 A network of HMMs.

language models in Section 4.7. Section 4.6 additionally proposes a method to construct complementary system that can improve the ASR performance when multiple systems are combined.

Certainly, SE is effective to improve ASR performance as shown in Chapter 3. But SE has side effects. Distortions caused by SE sometimes degrade ASR performance. Section 4.8 proposes to reduce the influence of speech distortions.

## 4.2 ASR system overview

### 4.2.1 GMM-HMM ASR systems

HMM is suitable for modeling non-stationary symbols that are composed of transient stationary states in short time. For example, paper [14, 148] shows examples of weather transition and ball pickup from a box including various colors of balls. Speech is essentially non-stationary because non-stationarity conveys the information. However, within the STFT window length (10 ms), the state is stationary. In each frame, speech is stationary and across frames states are changed and make up non-stationary speech. In addition to the (unrealistic) assumption of stationary states, it causes a discontinuity between states but they are not considered in this thesis. HMM is essentially a stationary method and it is difficult to directly model a time-changing features but this is widely used because of its simple principle.

HMM is a derivation of Markov model that satisfies Markov property. Markov property assumes the conditional probability of the current events only depends on the some previous events. In the case of ASR, Viterbi algorithm assumes that it only depends on the last events (first-order Markov state transition) and its assumption can extremely reduce the required comutation of HMMs [149]. It is important that one stationary signal source is active at any time and active source is probabilistically determined. Hidden means it is unknown which one is active.

Fig. 4.2 shows the simplest example of HMM that have two states and two arcs. This is composed of self loop and transition arc to the next state. When a symbol is input into the state

1, the probability is output corresponding the symbol. The state is transient and the transition probabilities are related to the arc between states. The state 1 is changed to the state 2 at the transition probability of $a_{12}$ and loops back at th probability of $a_{11} = 1 - a_{12}$ before waiting the next input.

These HMMs can be simply concatenated as in Fig. 4.3 to recognize arbitrary word sequences or sentence. This model has left-to-right transition from the initial state S to the final state E. This left-to-right structure corresponds to the time sequence. The HMM output of each state is not deterministic but depends on the probability distributions [16]. From the state $q_t$, the probability outputting observation vector $\boldsymbol{x}_t$ is $b_{q_t}(\boldsymbol{x}_t)$. Probability distributions of these types of signal sources are classified into discrete and continuous ones. Generally, continuous types is used and modeled as $M$ mixed distributions.

$$b_{q_t}(\boldsymbol{x}_t) = \sum_{m=1}^{M} c_{jk} \mathcal{N}(\boldsymbol{x}_t | \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}). \tag{4.1}$$

Here, $\boldsymbol{x}_t$ is an observation vector that can be modeled and $c_{jm}$ is a mixture weight for the $m$-th component of a mixture distribution at the state $j$. This mixture weight satisfies the condition:

$$\sum_{k=1}^{N_m} c_{jk} = 1, \quad c_{jk} \geq 0, \tag{4.2}$$

where $\boldsymbol{\mu}_{jm}$ is a mean vector and $\boldsymbol{\Sigma}_{jm}$ is a covariance matrix. $\mathcal{N}$ is usually Gaussian distribution.

Six parameters below are necessary to fix HMM completely.

(1) $N$: the number of states of a model

(2) $M$: the number of mixtures at each state

(3) $\boldsymbol{x}_{1:T}$: observation vector from 1 to $T$ frames ($\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$)

(4) $a_{ij}$ $(1 \leq i, j \leq N)$: state transition probability matrix

(5) $b_j(\boldsymbol{x}_t)$: probability distribution of an observation vector

(6) initial state distribution

Among them, $a_{ij}$ and $b_j$ are free parameters. The set of these parameters is annotated as model parameter $\lambda$ [14, 148].

If these three problems below can be solved, HMM is used for probability evaluation [148].

(1) when observation sequence and model are given, how to efficiently calculate a probability of observation sequence – probability evaluation –

(2) when observation sequence and model are given, how to select the optimal state sequences in a certain sense – the "optimal" state sequence –

(3) how to adjust model parameters that maximize a probability of observation sequence – parameter estimation –

HMM can be used to estimate probability because HMM is a generative model that generates non-stationary signals. When signal sequence $\boldsymbol{q} = \{q_1, q_2, \cdots, q_T\}$ is fixed, model $\lambda$ can calculate the probabilities $P(\boldsymbol{x}_{1:T}, \boldsymbol{q}|\lambda)$ for a concrete symbol sequence $\boldsymbol{x}_{1:T}$. This is because it is possible to calculate the probabilities for any non-stationary data sequences.

$$P(\boldsymbol{x}_{1:T}, \boldsymbol{q}|\lambda) = \pi_{q_1} \prod_{t=1}^{T} a_{q_t q_{t+1}} b_{q_t}(\boldsymbol{x}_t), \tag{4.3}$$

where $q_t$ is an active signal source at time $t$. $\pi_{q_1}$ is the probability when the signal source $q_1$ first activates. Summation of the probabilities of all sequences of possible signal sources,

$$P(\boldsymbol{x}_{1:T}|\lambda) = \sum_{q_1, q_2, \cdots, q_T} P(\boldsymbol{x}_{1:T}, \boldsymbol{q}|\lambda), \tag{4.4}$$

is the probability for a symbol sequence $\boldsymbol{x}_{1:T}$ with model $\lambda$. However, their combination numbers are exponentially increased and all combinations cannot be considered. This is a problem (1) and its efficient solution is Baum-Welch (Forward-Backward) algorithm. Forward variables $\alpha_t(j)$ are defined as

$$\alpha_t(j) = P(\boldsymbol{x}_{1:t}, q_t = j|\lambda) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(\boldsymbol{x}_t). \tag{4.5}$$

This is a probability of outputting partial observation sequence $\boldsymbol{x}_{1:t}$ until time$t$ and staying state $j$ at time $t$. Backward variables $\beta_t(j)$ are defined in the same way. This procedure calculates the probability $\gamma_t(j)$ of the state $j$ at time $t$.

$$\gamma_t(j) = P(q_t = j|\boldsymbol{x}_{1:t}, \lambda) = \frac{P(\boldsymbol{x}_{1:t}, q_t = j|\lambda)}{P(\boldsymbol{x}_{1:t}|\lambda)} = \frac{P(\boldsymbol{x}_{1:t}, q_t = j|\lambda)}{\sum_{i=1}^{N} P(\boldsymbol{x}_{1:t}, q_t = i|\lambda)}. \tag{4.6}$$

Here, because $P(\boldsymbol{x}_{1:T}, q_t = j|\lambda)$ is equivalent to $\alpha_t(j)\beta_t(j)$, the equation

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}, \tag{4.7}$$

holds. From $\gamma_t(i)$, at time $t$, the most probable state $q_t$ can be selected as

$$q_t^* = \arg\min_{1 \le i \le N} [\gamma_t(i)], \tag{4.8}$$

where arg min is an argument that maximizes a function [ ]. This Baum-Welch algorithm efficiently obtains probabilities for a symbol sequence.

For problem (2), there are more efficient algorithm because this needs only the optimal path. Baum-Welch algorithm calculates all possible paths including paths that have low probabilities. However, generally, total likelihood heavily depends on the paths that have the maximum probabilities. Instead of the summation in Eq. (4.5), Viterbi algorithm [8] only uses the maximum probability. This algorithm can always obtain the optimal path.

The HMM parameters are trainable. HMM training can construct probable models by local optimization of the likelihoods for training data from the initial distributions. This generally

called as expectation maximization (EM) algorithm [150]. This algorithm can be used for the parameter estimation and problem (3) can be solved. This EM algorithm has two advantages against other algorithms [151]:

(1) monotonic increase in an evaluation function (likelihood)

(2) easy implementation

Re-estimation of HMM parameters is as follows. The probability of the case where it exists at the state $i$ and time $t$ and it exists at the state $j$ and time$t+1$ is defined as

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|\boldsymbol{x}_{1:t}, \lambda). \tag{4.9}$$

$\gamma$ is calculated as

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). \tag{4.10}$$

Thus, transition counts can be related to the $\gamma$ and $\xi$ as

$$\begin{aligned} &\sum_{t=1}^{T-1} \gamma_t(i) = \text{Expectation of transition counts from the state } i \text{ at } \boldsymbol{x}_{1:T}, \\ &\sum_{t=1}^{T-1} \xi_t(i,j) = \text{Expectation of transition counts from the state } i \text{ to the state } j \text{ at } \boldsymbol{x}_{1:T}. \end{aligned} \tag{4.11}$$

For example, when each source obeys one single normal distribution, models with a initial model parameters are given and time sequence probabilistically generated by this model can be obtained.

Based on this, Viterbi path search can determine the maximum probable path and state transitions. From these, because it can determine the active source at each time (Viterbi segmentation), for each time mean and variance can be calculated and state transition probabilities can be calculated as a inverse of their durations. These parameters are substituted to the models and the repetition of this step increases the likelihood of probability models. The model accuracy and efficiency of training are two contradictory requirements. It is widely known that models that are overly tuned on the closed training data cannot perform well for the other open data. On the other hand, when free degree of model parameters is too low, the model cannot reflect training data fully [152]. It is very important to design training data.

### 4.2.2 DNN-HMM hybrid ASR systems

DNN-HMM hybrid ASR systems have been shown to outperform conventional GMM-HMM systems in a wide variety of conditions [137, 153]. Let us assume that DNN acoustic parameters $\theta$ are composed of $L$ hidden layer. Here, 0-th layer is the input layer and $(L+1)$-th layer is the output layer. For the $l$-th layer of DNN acoustic models ($0 \leq l \leq L+1$), $n$-dimensional input feature is denoted as $\boldsymbol{x}^l$. The output feature is $m$-dimensional and also an input feature

of the $(l+1)$-th layer, thus, this can be denoted as $\boldsymbol{x}^{l+1}$. Non-linear operation $\mathcal{T}$ is used in addition to linear operation. For hidden layers, sigmoid function is used as $\mathcal{T}$, whereas for the last layer, softmax function is used. Weight matrix of $\boldsymbol{A}^l_{m\times n}$ and offset $\boldsymbol{b}^l$ are trained using back propagation (fine-tuning) with stochastic gradient descent where the lower-suffix of the matrices represents their dimension. From the lower layer to the higher layer, feature $\boldsymbol{x}$ is propagated as

$$\boldsymbol{x}^{l+1} = \mathcal{T}\left(\boldsymbol{A}^l_{m\times n}\boldsymbol{x}^l + \boldsymbol{b}^l\right). \tag{4.12}$$

There are two types of DNN initializations: layer-wise training that constructs DNN by training hidden layers one by one and restricted Boltzmann machine (RBM) based initialization.

The DNN model provides posterior probabilities for the HMM state $j$ at frame $t$. In the hybrid DNN approach, the pseudo acoustic likelihood $p$ is obtained as

$$p\left(\boldsymbol{x}^0_t|j\right) \propto \frac{p\left(j|\boldsymbol{x}^0_t\right)}{p_0(j)}, \tag{4.13}$$

where $p_0(j)$ is the prior probability calculated from the count of training data. DNN input feature $\boldsymbol{x}^0_t$ is a spliced feature $[\boldsymbol{x}_{t-T_s}, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t+T_s}]$ in contiguous $(2T_s+1)$ frames. The DNN output is an output probability of each context-dependent HMM state. A softmax activation function is used for the output layer

$$p(j|\boldsymbol{x}^0_t) = \frac{\exp a(j|\boldsymbol{x}^0_t)}{\sum_{j'}\exp a(j'|\boldsymbol{x}^0_t)}, \tag{4.14}$$

where $a$ is the pre-activation value of the output layer node $j$, as a function of the input $\boldsymbol{x}^0_t$ to the DNN.

### 4.2.3 Feature adaptations

#### 4.2.3.1 Simple adaptation

To normalize the large variations of features among speakers and noises, feature adaptation is effective for both GMM and DNN based acoustic models. Simple feature adaptation is a mean and variance normalization. Cepstrum mean normalization (CMN) and cepstrum mean and variance normalization (CMVN) [154, 155] are widely used. These techniques compensate mean and variance mismatch between speakers. Histogram equation (HEQ) [156] is a more advanced CMN and CMVN. CMN and CMVN have a few parameters such as mean or variance. HEQ models distribution of acoustic features as a histogram and equalizes histograms of training data and those of evaluation data. Maximum likelihood linear regression (MLLR) [140] is a model-based adaptation. This one maximizes the likelihood of GMMs by adjusting means and variances of GMMs.

#### 4.2.3.2 fMLLR-based adaptation

This section introduces fMLLR [139] with SAT [135]. Feature adaptation methods can improve ASR accuracies in noisy environments by adapting to unknown and changing noise conditions

[139, 141]. fMLLR types of feature adaptations maximize a likelihood $\mathcal{L}$ for normal distributions $\mathcal{N}$ of the $j$-th state and $m$-th mixture with the mean $\boldsymbol{\mu}_{jm}$ and covariance $\boldsymbol{\Sigma}_{jm}$ as

$$\mathcal{L}_{jm}(\boldsymbol{x}_t) = |\boldsymbol{A}| \mathcal{N}\left(\hat{\boldsymbol{x}}_t | \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right), \tag{4.15}$$

where $\boldsymbol{x}_t$ is an observation at frame $t$ and $\hat{\boldsymbol{x}}_t$ is a transformed feature as

$$\hat{\boldsymbol{x}}_t \triangleq \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{b} = \boldsymbol{A}' \begin{bmatrix} \boldsymbol{x}_t \\ 1 \end{bmatrix}. \tag{4.16}$$

After adaptation, transformed features $\hat{\boldsymbol{x}}_t$ can be input in the same manner as the original feature.

#### 4.2.3.3 fMLLR for filter bank features

Conventional fMLLR is applied for DNN [157, 158, 159] after ASR decoding is performed using GMM. However, widely-used filter bank (FBANK) features cannot be represented well by a diagonal covariance GMM [157]. For this limitation, fMLLR with FBANK features did not improve the ASR performance [160] and it is necessary to de-correlate FBANK features before adaptation. In the adaptation phase, a global MLLT [126], $\boldsymbol{M}$, is applied to de-correlate FBANK features, whereas in the decoding phase, an inverse MLLT $\boldsymbol{M}^{-1}$ is applied to de-correlated and adapted fMLLR features as

$$\hat{\boldsymbol{x}}_t = \boldsymbol{M}^{-1}\boldsymbol{A}' \begin{bmatrix} \boldsymbol{M}\boldsymbol{x}_t \\ 1 \end{bmatrix}. \tag{4.17}$$

#### 4.2.3.4 BN and VTLN features

Two types of additional feature transformations are investigated: BN features [161, 162] and VTLN [163, 164, 165, 166]. Before DNN prevailed, to combine neural networks with a conventional GMM, a tandem structure was used [167, 168]. This approach has been extended to DNN and its extension–the BN feature–is widely used because conventional GMM can be used for decoding and features can be easily adapted for these types of structures. The BN feature is a lower dimensional hidden-layer unit output. To extract BN features, DNN is trained to predict phoneme states when the hidden layer size is smaller than the input layer size.

VTLN is another type of speaker normalization technique. Among several VTLN methods, we employ a simple linear approximation approach [166]. To approximate usual VTLN warped features $\boldsymbol{x}_t^\alpha$ with different warping factors $\alpha$'s, linear VTLN uses linear transformations $\boldsymbol{A}^\alpha$ and offsets $\boldsymbol{b}^\alpha$, which map an original feature $\boldsymbol{x}_t$ to the warped feature $\boldsymbol{x}_t^\alpha$ as

$$\boldsymbol{x}_t^\alpha \approx \boldsymbol{A}^\alpha \boldsymbol{x}_t + \boldsymbol{b}^\alpha. \tag{4.18}$$

These parameters ($\boldsymbol{A}^\alpha$ and $\boldsymbol{b}^\alpha$) are obtained to minimize square errors

$$\boldsymbol{A}^\alpha, \boldsymbol{b}^\alpha \leftarrow \underset{\boldsymbol{A}^\alpha, \boldsymbol{b}^\alpha}{\arg\min} \left|\boldsymbol{x}_t^\alpha - \left(\boldsymbol{A}^\alpha \boldsymbol{x}_t + \boldsymbol{b}^\alpha\right)\right|^2, \tag{4.19}$$

by using a subset of training data.

## 4.3 Linear discriminant analysis (LDA) of acoustic features

Feature transformation with dimensionality reduction is usually the first step in the front-end pipeline for ASR. Such methods allow the use of long-context features that can consider the influence across multiple frames directly instead of using traditional delta features [169]. One of the simplest and widely used methods has been LDA [125], which maximizes the ratio of the between-class variance to the within-class variance, where the classes are typically derived from the context-dependent phoneme states. An advantage of LDA is that it provides a simple and efficient closed-form solution to estimate the transformation. One of its limitations is the assumption of equal covariance for the classes. To relax the constraint of equal covariance of LDA, heteroscedastic discriminant analysis (HDA) and heteroscedastic LDA (HLDA) have been proposed [170, 171].

Another limitation of LDA is the lack of explicit consideration of speech recognizer (decoder) outputs. The purpose of feature transformation is essentially to provide features appropriate for recognition. LDA aims to improve discriminability of features but standard LDA deals the same way with classes that are easy to distinguish for the recognizer as with classes that are difficult to classify (i.e., easy to confuse).

Owing to the recent progress in discriminative training methods, it is well known that a sequential discriminative training with recognizer error tendencies is effective for various conventional techniques such as acoustic modeling or feature space discriminative training. Maximum mutual information (MMI) criterion [144] or minimum phoneme error (MPE) criterion [146] are effective training criteria because they consider the patterns of error at the recognition level, in order to focus on distinguishing the most important states. During training, these methods typically employ extended-Baum-Welch (EBW) updates, where the sufficient statistics for model parameter estimation are based on functions of the posterior probabilities of the recognition word sequences. Feature transformation based on such methods can improve the ASR performance further.

Linear feature transformation generally consists of projection matrices and offset terms. LDA is a global (single region) linear projection with no offsets. In contrast, region dependent linear transformation (RDLT) [172, 173] first divides the feature space into regions, and for each region separate transforms can be applied. Discriminative approaches such as MPE-HLDA [174], which is an extension of HLDA based on the MPE criterion, feature space MPE (f-MPE) [175], and MMI-SPLICE [176], have been proposed. Such methods typically require iterative gradient-descent optimization. Typically, LDA features are still used as input to such methods since they are simple to compute and provide a reasonable starting point.

The proposed method is an extension of standard LDA based on the MMI objective function in order to consider the recognition posteriors when feature statistics are calculated. The advantages of the proposed method are the existence of a closed-form solution, and the simplicity of implementation which amounts to a simple modification of the sufficient statistics computation.

This section describes the conventional LDA [125], mainly from the perspective described

in [170, 177]. The next section describes our proposed MMI approach. Experiments on two different tasks show that the proposed method improves ASR performance on two data sets in Section 4.3.3.

### 4.3.1 Conventional maximum likelihood LDA

For LDA, $n$-dimensional input feature $\boldsymbol{x}_t \in \mathbb{R}^n$ is usually obtained by concatenating original MFCC features of contiguous several frames. LDA feature transformation [125, 177] transforms $\boldsymbol{x}_t$ to lower dimensional feature $\boldsymbol{y}_t \in \mathbb{R}^p$ as

$$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{x}_t, \tag{4.20}$$

where $\boldsymbol{A}$ is an LDA feature transformation matrix whose dimension is $p \times n$, which ($p < n$). The objective function of LDA is given as

$$\arg\max_{\boldsymbol{A}} \frac{|\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^\top|}{|\boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^\top|}, \tag{4.21}$$

where $\top$ denotes a transposition and $\boldsymbol{B}$ and $\boldsymbol{W}$ denote $n \times n$ between-class scatter matrix and within-class scatter matrix as defined in Eq. (4.22), respectively:

$$\begin{aligned} \boldsymbol{B} &= \frac{1}{\sum_j N_j} \sum_j N_j \boldsymbol{\mu}_j^{\mathrm{x}} (\boldsymbol{\mu}_j^{\mathrm{x}})^\top - \bar{\boldsymbol{\mu}}^{\mathrm{x}} (\bar{\boldsymbol{\mu}}^{\mathrm{x}})^\top, \\ \boldsymbol{W} &= \frac{1}{\sum_j N_j} \sum_j N_j \boldsymbol{\Sigma}_j^{\mathrm{x}}, \end{aligned} \tag{4.22}$$

where $\boldsymbol{\mu}^{\mathrm{x}}$ and $\boldsymbol{\Sigma}^{\mathrm{x}}$ are the mean vector and co-variance matrix in the original vector $\boldsymbol{x}$ space, $N_j$ is the count of elements which belong to the $j$-th class, and $\bar{\boldsymbol{\mu}}^{\mathrm{x}}$ is the average of all vectors $\boldsymbol{\mu}_j^{\mathrm{x}}$. Generally, $\boldsymbol{\mu}_j^{\mathrm{x}}$ and $\boldsymbol{\Sigma}_j^{\mathrm{x}}$ are computed [177] for class $j$ as

$$\begin{aligned} N_j &= \sum_t \psi_t(j), \\ \boldsymbol{\mu}^{\mathrm{x}} &= \frac{1}{N_j} \sum_t \psi_t(j) \boldsymbol{x}_t, \\ \boldsymbol{\Sigma}_j^{\mathrm{x}} &= \frac{1}{N_j} \sum_t \psi_t(j) \boldsymbol{x}_t \boldsymbol{x}_t^\top - \boldsymbol{\mu}_j^{\mathrm{x}} (\boldsymbol{\mu}_j^{\mathrm{x}})^\top, \end{aligned} \tag{4.23}$$

where, $\psi_t(j)$ are class membership weights relating $\boldsymbol{x}_t$ to class $j$. In the classic LDA, the class assignments, given by $j = l(t)$, are hard, so $\psi_t(j)$ can be defined as:

$$\psi_t(j) = \begin{cases} 1 & (l(t) = j), \\ 0 & (\text{otherwise}). \end{cases} \tag{4.24}$$

Here, we assume that LDA class $j$ is related to the HMM state number as in the most general case. In this case, alignments by the HMM model correspond to the class label.

A solution to LDA is obtained by solving the following generalized eigenvalue problem [178],

$$\boldsymbol{B}v = \lambda \boldsymbol{W}v, \tag{4.25}$$

and assigning to the rows of $\boldsymbol{A}$ the eigenvectors $v_{1:p}^T$ corresponding to the $p$ largest eigenvalues $\lambda_{1:p}$.

It has been shown by Kumar *et al.* that standard LDA has the same optimum as a maximum likelihood problem [170]. In this problem, the model has tied state-dependent variances in $y_t = \boldsymbol{A}x_t$, and the mean and variance in the orthogonal subspace $y'_t = \boldsymbol{A}'x_t$ are state-independent, where $\boldsymbol{A}'$ is an $(n-p) \times n$ matrix having rows orthogonal to those of $\boldsymbol{A}$.

This result can be generalized to a full HMM model, with tied parameters in the style of Kumar, by considering the maximum likelihood objective function:

$$\mathcal{F}^{\mathrm{MLK}} = \ln P(\omega_r, \boldsymbol{Y}), \tag{4.26}$$

where $\boldsymbol{Y} = \{\boldsymbol{y}_1, \cdots\}$ is the sequence of transformed feature vectors, and $\omega_r$ is the correct word label. The derivative of this model with respect to a model parameter $\theta_j$ is

$$\frac{\partial \mathcal{F}^{\mathrm{MLK}}}{\partial \theta_j} = \sum_t \sum_j \frac{\partial \mathcal{F}^{\mathrm{MLK}}}{\partial \ln p(\boldsymbol{y}_t|j)} \frac{\partial \ln p(\boldsymbol{y}_t|j)}{\partial \theta_j} = \sum_t \sum_j \gamma_t(j) \frac{\partial \ln p(\boldsymbol{y}_t|j)}{\partial \theta_j}, \tag{4.27}$$

where $p(\boldsymbol{y}_t|j)$ is the acoustic model state conditional probability. Setting these derivatives equal to 0 and solving for model parameters leads to the state-dependent means and variances as calculated in Eq. (4.23) with

$$\psi_t(j) = \gamma_t^{\mathrm{num}}(j), \tag{4.28}$$

for state $j$. This again leads to a solution to the LDA problem using the generalized eigenvalue problem (4.25), this time with soft class membership determined by the state posteriors. For models estimated using the Baum-Welch algorithm, the above LDA statistics more closely correspond to those used in estimating the model. This means that the matrices $\boldsymbol{B}$ and $\boldsymbol{W}$ are more accurately estimated in this case.

### 4.3.2 Sequential maximum mutual information LDA

#### 4.3.2.1 Derivation from MMI objective function

The previous section describes the linear transformation which maximizes scatter between classes and minimizes scatter within classes based on the correct labels as in Eq. (4.21). However, because the maximum likelihood statistics are different from the MMI statistics, the resulting $\boldsymbol{B}$ and $\boldsymbol{W}$ are not accurate for MMI-based models. Similar to the MMI discriminative training of acoustic model parameters, posteriors of denominator lattices $\gamma_t^{\mathrm{den}}$ should be taken into account. We call this method sequential MMI LDA (sLDA).

The MMI objective function is given as

$$\mathcal{F}^{\mathrm{MMI}} = \ln \frac{P(\omega_r, \boldsymbol{Y})}{\sum_\omega P(\omega, \boldsymbol{Y})}, \tag{4.29}$$

where $\omega$ are the hypotheses of the original system. The derivative of the MMI objective function by state-dependent model parameters $\theta_j$, as in MMI-SPLICE [176], is

$$\frac{\partial \mathcal{F}^{\mathrm{MMI}}}{\partial \theta_j} = \sum_t \sum_j \frac{\partial \mathcal{F}^{\mathrm{MMI}}}{\partial \ln p(\boldsymbol{y}_t|j)} \frac{\partial \ln p(\boldsymbol{y}_t|j)}{\partial \theta_j} = \sum_t \sum_j \Delta_t(j) \frac{\partial \ln p(\boldsymbol{y}_t|j)}{\partial \theta_j}, \tag{4.30}$$

where $p(\boldsymbol{y}_t|j)$ is the acoustic model state conditional probability. This leads to the same mean and variance estimation as Eq. (4.23), except that here $\psi_t(j) = \Delta_t(l(t))$. However, since $\Delta_t(j)$ can be negative, usually extended Baum-Welch updates are used, because they maintain positivity. Here we introduce a parameter $\alpha$ ($0 \leq \alpha \leq 1$) that reduces the strength of the denominator term $\gamma_t^{\mathrm{den}}(j)$:

$$\psi_t(j) = \gamma_t^{\mathrm{num}}(j) - \alpha \gamma_t^{\mathrm{den}}(j). \tag{4.31}$$

If $\alpha$ equals to zero, this equation reduces to that of LDA.

The proposed method can be interpreted as a form of LDA with a soft feature selection [179] corresponding closely to the MMI model. Little weight is imposed on the data where $\gamma_t^{\mathrm{den}}(j)$ is near one and this corresponds to the correct case for a recognizer. This realizes an adjustment of the weight of the training data according to the errors made by the recognizer. However, as the between-class variance $\boldsymbol{B}$ remains global, it is only slightly affected by the MMI-based weights, and this method still focuses on all classes. Nevertheless, it has a simple closed-form solution, and an easy implementation, so may be useful as a starting point for more advanced discriminative transforms.

#### 4.3.2.2 I-smoothing interpretation

Equation (4.31) can be rewritten as

$$\psi_t(j) = (1-\alpha)\gamma_t^{\mathrm{num}}(j) + \alpha \Delta_t(j). \tag{4.32}$$

This equation can be interpreted as a smoothing between the difference statistics $\Delta_t(j)$ and the class label posterior $\gamma_t^{\mathrm{num}}(j)$ with interpolation ratio $\alpha$. Thus, by setting the parameter $\alpha$ less than 1, $\alpha$ helps avoid over-training and is related to I-smoothing [146], which is widely used for discriminative training of acoustic models.

#### 4.3.2.3 Boosted MMI extension

In analogy to boosted MMI [175], we can introduce a boosting factor $b$ that boosts the posteriors of hypotheses based on the phoneme accuracy. The boosted MMI objective function is:

$$\mathcal{F}^{\mathrm{bMMI}} = \ln \frac{P(\omega_r, \boldsymbol{Y})}{\sum_\omega P(\omega, \boldsymbol{Y}) e^{-bH(\omega, \omega_r)}}, \tag{4.33}$$

where $H(\omega, \omega_r)$ is the phoneme accuracy. The boosted version of the weights can be obtained as in the classical boosted MMI framework, by using the forward-backward algorithm on the

denominator lattice, and adding, for each state, $-b$ times the contribution to the sentence level accuracy. Denoting by $\gamma_t^{b,\mathrm{den}}(j)$ the denominator term, we obtain a bMMI version of the weights $\psi_t^b(j)$:

$$\psi_t^b(j) = \gamma_t^{\mathrm{num}}(j) - \alpha\gamma_t^{b,\mathrm{den}}(j). \tag{4.34}$$

The boosting factor $b$ is typically taken to be negative so as to put more focus on frames with low accuracy than on those with high accuracy.

### 4.3.3 Experimental setups

We evaluated the performance improvement on two corpora: the Corpus of Spontaneous Japanese (CSJ) [180] and the second CHiME challenge Track 2 (details are in Section 5.2). The former is one of the most widely used large-vocabulary continuous speech recognition (LVCSR) tasks (vocabulary size is about 70k). Three types of test sets are provided and each set consists of 10 speakers' lecture-style speech. Test sets 1, 2, and 3 contain 22,682, 23,226, and 14,896 words, respectively. The first aim of our experiments is to validate the effectiveness of the proposed sLDA compared to the conventional LDA when changing the parameters $\alpha$ and $b$ in Eq. (4.34). The HMM was trained with maximum likelihood estimation using 0th∼12th order MFCCs + $\Delta$ + $\Delta\Delta$, the number of context-dependent HMM states was 3,500 and the total number of Gaussians was 96,000.

The second CHiME challenge task is aimed to validate the performance of the proposed sLDA for noise robust speech recognition task, and the effectiveness of its combinations with discriminative training of acoustic models (GMM and DNN) and f-bMMI. We used noise-suppressed single-channel data obtained by prior-based binary masking (Section 5.2.2). The number of HMM states was 2,500 and the total number of Gaussians was 15,000. For the DNN, we used the nnet2 implementation of DNN training in Kaldi with 3 hidden layers and 1,000,000 parameters. The initial learning rate was 0.01 and was decreased to 0.001 at the end of training. The baseline features were 0th∼12th order MFCCs + $\Delta$ + $\Delta\Delta$. Moreover, we combine LDA with MLLT. For the CHiME corpus, SAT (Section 4.2.3.2) were also applied.

### 4.3.4 Results and discussion

#### 4.3.4.1 CSJ (LVCSR)

Table 4.1 shows the experimental results on the CSJ corpus. Although the performance improvement depended on the parameter $\alpha$, overall, the proposed sLDA worked better than the conventional LDA ($\alpha = 0$) even when combined with MLLT. For the best case (bold case in the table), absolute 0.21% and 0.19% WER reductions for sLDA and sLDA+MLLT respectively were observed. Unfortunately, the boosted extension had little impact on the results, and for the rest of the experiments, the boosting factor $b$ was set to zero.

Table 4.1 WER of the conventional LDA ($\alpha = 0$) and the proposed sequential maximum mutual information LDA (sLDA) with different $\alpha$ and $b$, which are smoothing and boosting factors in Eq. (4.34), respectively, on CSJ database.

| | $\alpha$ | $b$ | test1 | test2 | test3 | Avg. |
|---|---|---|---|---|---|---|
| LDA | 0 | 0 | 20.42 | 17.95 | 19.22 | 19.20 |
| | 0.1 | 0 | **20.39** | 17.81 | 19.49 | 19.23 |
| | 0.3 | 0 | 20.47 | 17.93 | 19.28 | 19.23 |
| | 0.5 | 0 | 20.44 | 17.81 | 19.14 | 19.13 |
| | 0.7 | 0 | 20.40 | 17.83 | 19.03 | 19.09 |
| | 1.0 | 0 | 20.51 | **17.68** | **18.77** | **18.99** |
| | 0.1 | −0.1 | 20.46 | 17.86 | 19.29 | 19.20 |
| | 0.3 | −0.1 | 20.28 | 17.74 | 19.21 | 19.08 |
| | 0.5 | −0.1 | 20.38 | 17.87 | 19.08 | 19.11 |
| | 0.7 | −0.1 | 20.43 | 17.63 | 19.13 | 19.06 |
| | 1.0 | −0.1 | 20.60 | 17.65 | 18.91 | 19.05 |
| LDA | 0 | 0 | 19.09 | 16.31 | 17.21 | 17.54 |
| +MLLT | 0.1 | 0 | 19.13 | 15.96 | 17.23 | 17.44 |
| | 0.3 | 0 | 19.08 | **15.91** | 17.07 | **17.35** |
| | 0.5 | 0 | 19.04 | 16.12 | 17.25 | 17.47 |
| | 0.7 | 0 | 19.09 | 16.03 | 17.11 | 17.41 |
| | 1.0 | 0 | 18.90 | 16.24 | **16.94** | 17.36 |
| | 0.1 | −0.1 | 19.20 | 16.21 | 17.33 | 17.58 |
| | 0.3 | −0.1 | 19.07 | 16.21 | 17.09 | 17.46 |
| | 0.5 | −0.1 | 18.96 | 16.11 | 17.07 | 17.38 |
| | 0.7 | −0.1 | **18.87** | 16.09 | 17.19 | 17.38 |
| | 1.0 | −0.1 | 19.17 | 16.05 | 17.11 | 17.44 |

Table 4.2 WER[%] for isolated speech (**si_dt_05**) of the CHiME challenge with different $\alpha$s using ML acoustic model for noisy speech recognition with noise suppression by prior-based binary masking (sLDA+MLLT).

| $\alpha$ | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| 0 | 64.64 | 54.24 | 46.35 | 37.91 | 32.75 | 28.96 | 44.14 |
| 0.1 | 64.64 | 53.81 | 46.45 | 38.65 | 32.75 | 29.15 | 44.24 |
| 0.3 | 64.88 | **53.72** | 45.58 | **37.13** | **31.89** | **28.43** | **43.61** |
| 0.5 | 64.71 | 53.84 | 46.20 | 37.81 | 32.25 | 28.81 | 43.94 |
| 0.7 | 64.48 | 54.43 | 45.88 | 37.51 | 32.44 | 28.69 | 43.91 |
| 1.0 | **64.36** | 54.29 | **45.01** | 37.81 | 32.59 | 28.96 | 43.84 |

#### 4.3.4.2 Second CHiME Challenge Track 2 (Noise robust ASR)

Table 4.2 continues to investigates further the influence of the parameter $\alpha$ on performance through experiments on the CHiME challenge Track 2. MLLT is used in addition to the proposed sLDA. In average, for the cases where $\alpha$ is 0.3 or more, the speech recognition performance was improved and the case $\alpha = 0.3$ achieved the best improvement (0.53% absolute WER reduction), which is the same as in Table 4.1. From Tables 4.1 and 4.2, we validate that the proposed LDA was superior to the conventional LDA on two different ASR tasks.

Table 4.3 shows the results with discriminative training of acoustic model (bMMI) and feature

Table 4.3 WER[%] for isolated speech (**si_dt_05**) using ML and discriminatively trained acoustic model (bMMI) with feature-space discriminative training (f-bMMI). LDA+MLLT (upper), sLDA+MLLT (lower).

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| ML | 64.64 | 54.24 | 46.35 | 37.91 | 32.75 | 28.96 | 44.14 |
| bMMI | 63.39 | 52.54 | 44.56 | 35.60 | 30.98 | 28.10 | 42.53 |
| f-bMMI | 60.92 | 50.41 | 41.76 | 33.59 | 29.56 | 25.90 | 40.36 |
| ML | 64.88 | 53.72 | 45.58 | 37.13 | 31.89 | 28.43 | 43.61 |
| bMMI | 62.75 | 51.78 | 44.24 | 35.92 | 30.80 | 27.32 | 42.14 |
| f-bMMI | **60.27** | **49.26** | **41.08** | **32.95** | **28.63** | **25.17** | **39.56** |

Table 4.4 WER[%] for isolated speech (**si_dt_05**) with speaker adaptive training, speaker adaptation (fMLLR), and minimum Bayes risk decoding (MBR). LDA+MLLT (upper), sLDA+MLLT (lower).

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| ML | 59.94 | 47.93 | 39.83 | 33.01 | 28.00 | 23.47 | 38.70 |
| bMMI | 56.90 | 45.79 | 37.60 | 30.31 | 26.15 | 21.74 | 36.42 |
| f-bMMI | 52.93 | 42.62 | 34.59 | 27.63 | 24.27 | 20.24 | 33.71 |
| (+MBR) | 52.65 | 42.04 | 33.75 | 27.05 | 23.74 | 19.91 | 33.19 |
| DNN | 52.78 | 42.50 | 34.08 | 27.05 | 24.13 | 20.12 | 33.44 |
| bMMI | 47.34 | 36.33 | 28.96 | 23.40 | 20.03 | 17.05 | 28.85 |
| (+MBR) | 46.79 | 35.68 | 28.44 | 22.88 | 19.91 | 16.64 | 28.39 |
| ML | 59.21 | 48.40 | 39.28 | 32.41 | 27.72 | 22.86 | 38.31 |
| bMMI | 56.14 | 45.51 | 36.69 | 29.55 | 26.08 | 21.33 | 35.88 |
| f-bMMI | 53.09 | 43.34 | 33.71 | 27.16 | 23.93 | 19.78 | 33.50 |
| (+MBR) | 52.60 | 42.51 | 33.03 | 26.38 | 23.34 | 19.18 | 32.84 |
| DNN | 52.91 | 41.81 | 32.56 | 27.73 | 24.31 | 19.68 | 33.17 |
| bMMI | 47.31 | 36.13 | 28.49 | 23.50 | 20.00 | 16.57 | 28.67 |
| (+MBR) | **46.59** | **35.31** | **27.84** | **22.82** | **19.69** | **16.49** | **28.12** |

space discriminative training (f-bMMI). For both cases, the proposed method improved the speech recognition performance, especially for the f-bMMI case (0.8% absolute WER reduction). The combination of the proposed method and f-bMMI achieved an additional improvement. This suggests that preliminary discriminative classification of the proposed method provided a good initialization to f-bMMI, which is also discriminative feature transformation with more precise region-dependent modeling.

Tables 4.4 and 4.5 show the results on the development and evaluation sets additionally with speaker adaptive training, fMLLR type speaker adaptation, and DNN system in order to validate the effectiveness of the proposed method in a state-of-the-art ASR system. Although for the DNN system the average ASR performance degraded on the evaluation set, the proposed method improved the performance for all the SNR conditions in the development set, and for half of the SNR conditions (−3, 3, and 9dB) in the evaluation set. Overall, the proposed method improved the average ASR performance by up to 0.9% absolute.

Table 4.5 WER[%] for isolated speech (**si_et_05**) with speaker adaptive training and speaker adaptation (fMLLR). LDA+MLLT (upper), sLDA+MLLT (lower). In this table, DNN is DNN with boosted MMI.

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| ML | 50.91 | 41.64 | 33.89 | 26.30 | 21.61 | 18.85 | 32.20 |
| f-bMMI | 44.54 | 35.91 | 29.24 | 22.31 | 17.77 | 15.88 | 27.61 |
| (+MBR) | 44.51 | 35.42 | 28.81 | 21.46 | 17.41 | 14.98 | 27.10 |
| DNN | 37.98 | 28.26 | 21.86 | 17.71 | 12.61 | 11.75 | 21.70 |
| (+MBR) | **37.14** | 27.35 | **21.41** | 16.94 | **12.55** | 11.54 | **21.16** |
| ML | 50.46 | 42.05 | 32.80 | 26.42 | 21.22 | 18.61 | 31.93 |
| f-bMMI | 44.85 | 35.05 | 27.69 | 21.43 | 17.34 | 14.74 | 26.85 |
| (+MBR) | 44.07 | 34.09 | 27.22 | 20.33 | 16.85 | 14.61 | 26.20 |
| DNN | 38.63 | 27.54 | 22.55 | 17.37 | 13.23 | 11.69 | 21.84 |
| (+MBR) | 37.98 | **27.16** | 21.73 | **16.93** | 12.83 | **11.23** | 21.31 |

### 4.3.5 Conclusion

This section proposed to extend LDA based on sequential MMI training methods by using the discriminatively modified sufficient statistics computed from the lattices. The advantages of the proposed method are its low complexity and ease of implementation, in that it boils down to a simple modification of the computation of the sufficient statistics. Experiments on both an LVCSR task and a noise robust ASR task show its effectiveness. Although our approach is based on the closed-form solution of a generalized eigenvalue problem and is in that regard different from other discriminative feature transformation methods based on EBW or gradient-descent optimization techniques, future work will consider in more depth the theoretical relationships between them.

## 4.4 Discriminative training methods of acoustic models

### 4.4.1 Cross-entropy (CE) training of DNNs

For the CE criterion, the objective function is

$$\mathcal{F}_{\mathrm{CE}}(\theta) = \sum_r \sum_t \sum_j \hat{p}(j,t) \ln \frac{\hat{p}(j,t)}{p(j|\boldsymbol{x}_t^0)}, \tag{4.35}$$

where $\hat{p}(j,t)$ is the reference distribution for class label $j$ at time $t$. The gradient with respect to $a$ is

$$\frac{\partial \mathcal{F}_{\mathrm{CE}}}{\partial a(j)} = p(j|\boldsymbol{x}_t^0) - \hat{p}(j,t). \tag{4.36}$$

Gradient descent based on the chain rule, known as back-propagation, can then be used for optimization of the DNN parameters

### 4.4.2 MMI discriminative training

The goal of discriminative training algorithms is to obtain models that minimize the empirical risk computed from the correct labels and recognition hypotheses. Several training criteria have been introduced [145, 181], such as MMI [144], minimum classification error [147], or MPE [146]. We focus on MMI in this work, because MMI is the most widely used criterion and because it is the starting point for the more advanced bMMI, which we use below.

The goal of MMI training is to maximize the mutual information between correct labels and recognition hypotheses, based on the following objective function:

$$\mathcal{F}_{\mathrm{MMI}}(\lambda) = \sum_r \ln \frac{P_\lambda\left(s^{(r)}, \boldsymbol{X}\right)}{\sum_s P_\lambda\left(s, \boldsymbol{X}\right)} = \sum_r \ln \frac{p_\lambda\left(\boldsymbol{x}^{(r)}|\mathcal{H}_{s^{(r)}}\right)^\kappa p_L(s^{(r)})}{\sum_s p_\lambda\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)^\kappa p_L(s)}, \tag{4.37}$$

where $\boldsymbol{x}^{(r)} = (\boldsymbol{x}_1^{(r)}, \ldots, \boldsymbol{x}_t^{(r)}, \ldots, \boldsymbol{x}_{T_r}^{(r)})$ is the $r$-th utterance's feature sequence of length $T_r$; The product of the acoustic model score $p_\lambda$ and the language model score $p_L$ is denoted by $P_\lambda$. $\lambda$ denotes the GMM-based acoustic model parameters composed of mixture weights, mean vectors, and (diagonal) covariance matrices; these parameters are optimized using the extended Baum-Welch algorithm; $\mathcal{H}_{s^{(r)}}$ and $\mathcal{H}_s$ are the HMM sequences that represent the correct label $s^{(r)}$ and a recognition result $s$, respectively; $p_\lambda$ is the acoustic model likelihood, $\kappa$ is the acoustic scale, and $p_L$ is the language model likelihood.

While MMI is effective, performance can be further improved by giving more weight to the training data that is improperly recognized, as proposed in the bMMI framework [175]. The above objective function is extended to a boosted version as follows:

$$\mathcal{F}_{\mathrm{bMMI}}(\lambda) = \sum_r \ln \frac{p_\lambda\left(\boldsymbol{x}^{(r)}|\mathcal{H}_{s^{(r)}}\right)^\kappa p_L(s^{(r)})}{\sum_s p_\lambda\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)^\kappa p_L(s) \mathrm{e}^{-bA(s,s^{(r)})}}, \tag{4.38}$$

where $A(s, s^{(r)})$ is the phoneme accuracy of hypothesis $s$ for a reference $s^{(r)}$, and $b \geq 0$ is a boosting factor that controls the phoneme accuracy dependent weight. In this section, we study

the performances of MMI and bMMI for noisy speech ASR, comparing them to the performance of ML.

### 4.4.3 MMI discriminative training of GMMs

In GMM training, Eq. (4.37) is broken down into the update formulae for the mean $\boldsymbol{\mu}_{jm}$ and covariance $\boldsymbol{\Sigma}_{jm}$ of GMM (HMM state $j$ and Gaussian index $m$) as

$$\begin{aligned}\boldsymbol{\mu}'_{jm} &= \frac{\sum_t \Delta_{jm,t}\boldsymbol{x}_t + D_{jm}\boldsymbol{\mu}_{jm}}{\sum_t \Delta_{jm,t} + D_{jm}}, \\ \boldsymbol{\Sigma}'_{jm} &= \frac{\sum_t \Delta_{jm,t}\boldsymbol{x}_t\boldsymbol{x}_t^\top + D_{jm}(\boldsymbol{\Sigma}_{jm} + \boldsymbol{U}_{jm})}{\sum_t \Delta_{jm,t} + D_{jm}} - \boldsymbol{U}'_{jm},\end{aligned} \tag{4.39}$$

where $\Delta_{jm,t}$ is $\gamma^{num}_{jm,t} - \gamma^{den}_{jm,t}$, $\gamma^{num}_{jm,t}$ and $\gamma^{den}_{jm,t}$ are the numerator and denominator of the posteriors of Eq. (4.37) or (4.38), and $\top$ denotes the transpose. $\boldsymbol{U}_{jm}$ and $\boldsymbol{U}'_{jm}$ denote $\boldsymbol{\mu}_{jm}\boldsymbol{\mu}_{jm}^\top$ and $\boldsymbol{\mu}'_{jm}\boldsymbol{\mu}'^\top_{jm}$, respectively. These update formulae are introduced by approximating the update formulae for discrete HMM optimization [182]. The Gaussian-specific learning rate $D_{jm}$ is set to make $\Sigma'_{jm}$ positive definite. The mixture weights $\pi_{jm}$ of GMM can be also optimized [175].

### 4.4.4 MMI discriminative training of DNNs

GMM-HMM systems have constituted the mainstream architecture for decades, but DNN-HMM hybrid systems have outperformed them in recent years when used in clean speech conditions. In this section, we investigate the effectiveness of DNN-HMM hybrid systems in noisy and reverberant speech conditions, and we show that these systems can bring further improvements compared to our challenge submission system [183]. In particular, we explore the benefits of sequence-level discriminative training methods for DNNs. DNNs are already discriminative at the frame level, because they are constructed based on discriminative criteria such as CE. Sequence-level discriminative training goes further in that it attempts to minimize the risk on the whole sequence instead of independently on each single frame; this type of training has been shown to improve performance over simple cross-entropy training [184, 185].

A DNN model with parameters $\theta$ outputs posterior probabilities $p_\theta(j|\boldsymbol{x}_t^{(r)})$ for each HMM state $j$ at frame $t$. These probabilities are computed using a softmax layer applied to the top layer of the DNN:

$$p_\theta(j|\boldsymbol{x}_t^{(r)}) = \frac{\exp a_\theta(j|\boldsymbol{x}_t^{(r)})}{\sum_{j'} \exp a_\theta(j'|\boldsymbol{x}_t^{(r)})}, \tag{4.40}$$

where $a_\theta$ is the output of the top layer. Each layer of the DNN transforms the outputs of the previous layer through an affine transform, whose parameters are a subset of $\theta$, followed by a non-linear operation such as a sigmoid.

In order to use the classical HMM-based decoding framework, hybrid DNN-HMM systems replace the acoustic likelihood of GMMs by a pseudo-likelihood $p_\theta\left(\boldsymbol{x}_t^{(r)}|j\right)$ obtained as

$$p_\theta\left(\boldsymbol{x}_t^{(r)}|j\right) \propto \frac{p_\theta\left(j|\boldsymbol{x}_t^{(r)}\right)}{p_0(j)}, \tag{4.41}$$

where $p_0(j)$ is the prior probability calculated from the count of states in the training data.

The values of the parameters $\theta$ are trained discriminatively according to the MMI criterion. The (boosted) MMI objective function is similar to that shown in Eqs. (4.37) and (4.38); the only difference is that the GMM likelihoods $p_\lambda\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)$ are replaced for the whole sequence by the equivalent DNN pseudo-likelihoods $p_\theta\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)$:

$$\mathcal{F}_{\text{MMI}}(\theta) = \sum_r \ln \frac{p_\theta\left(\boldsymbol{x}^{(r)}|\mathcal{H}_{s^{(r)}}\right)^\kappa p_L(s^{(r)})}{\sum_s p_\theta\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)^\kappa p_L(s)}. \tag{4.42}$$

The boosted version of Eq. (4.42) is:

$$\mathcal{F}_{\text{bMMI}}(\theta) = \sum_r \ln \frac{p_\theta\left(\boldsymbol{x}^{(r)}|\mathcal{H}_{s^{(r)}}\right)^\kappa p_L(s^{(r)})}{\sum_s p_\theta\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)^\kappa p_L(s)\mathrm{e}^{-bA(s,s^{(r)})}}. \tag{4.43}$$

The gradient of the objective function with respect to the top layer output $a_\theta$ can be obtained by the chain rule as:

$$\frac{\partial \mathcal{F}_{\text{bMMI}}(\theta)}{\partial a_\theta(j)} = \sum_{j'} \frac{\partial \mathcal{F}_{\text{bMMI}}}{\partial \ln p_\theta\left(\boldsymbol{x}^{(r)}|j'\right)} \frac{\partial \ln p_\theta\left(\boldsymbol{x}^{(r)}|j'\right)}{\partial a_\theta(j)} = \kappa(\gamma_{j,t}^{\text{num}} - \gamma_{j,t}^{\text{den}}), \tag{4.44}$$

where $\gamma_{j,t}^{\text{num}}$ and $\gamma_{j,t}^{\text{den}}$ are the posteriors of state $j$ at frame $t$ in the numerator and denominator of (4.43) (and similarly for Eq. (4.42)). The efficient calculation of these quantities is a classical step of MMI and MPE derivations for GMM systems and is described in detail in [146, 185]. All of the DNN parameters are estimated using the back-propagation procedure that begins with Eq. (4.44).

### 4.4.5 sMBR discriminative training of DNNs

The parameters $\theta$ are trained discriminatively according to the sequence-level minimum Bayes risk (sMBR) criterion:

$$\mathcal{F}_{\text{sMBR}}(\theta) = \sum_r \frac{\sum_s p_\theta\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)^\kappa p_L(s) A(s, s^{(r)})}{\sum_s p_\theta\left(\boldsymbol{x}^{(r)}|\mathcal{H}_s\right)^\kappa p_L(s)}, \tag{4.45}$$

where $A$ is the raw frame accuracy. The gradient of the objective function with respect to $a_\theta$ can be obtained as

$$\frac{\partial \mathcal{F}_{\text{sMBR}}(\theta)}{\partial a_\theta(j)} = \sum_{j'} \frac{\partial \mathcal{F}_{\text{sMBR}}(\theta)}{\partial \ln p_\theta\left(\boldsymbol{x}^{(r)}|j'\right)} \frac{\partial \ln p_\theta\left(\boldsymbol{x}^{(r)}|j'\right)}{\partial a_\theta(j)} = \kappa\gamma_{j,t}\left(\hat{A}(j) - \hat{A}\right), \tag{4.46}$$

where $\hat{A}(j)$ is the average accuracy of all hypotheses in the lattice whose state at frame $t$ is $j$; $\hat{A}$ is the average accuracy of all hypotheses; and $\gamma_{j,t}$ is the posteriors of state $j$ for all hypotheses in the lattice. The back-propagation procedure with Eq. (4.46) updates $\theta$.

### 4.4.6 Feature-space MMI discriminative training

In addition to the acoustic model, sequence discriminative training can also be used to derive a feature transformation. This is referred to as feature-space discriminative training [136]. In this section, the $I$-dimensional vector $\boldsymbol{x}_t \in \mathbb{R}^I$ denotes the original static features without dynamic features (that is $\boldsymbol{x}_t$ does not include $\Delta$ and $\Delta\Delta$; this is unlike the previous sections). The transformed features $\boldsymbol{y}_t \in \mathbb{R}^I$ are obtained by adding $\boldsymbol{x}_t$ to an offset determined by applying a linear transformation $\boldsymbol{M}$ to a high-dimensional feature vector $\boldsymbol{h}_t \in \mathbb{R}^J$, where $\boldsymbol{M}$ is estimated using sequence discriminative training:

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{M}\boldsymbol{h}_t. \tag{4.47}$$

The dimension $J$ of $\boldsymbol{h}_t$ is assumed to be much larger than the dimension $I$ of the original features $\boldsymbol{x}_t$ (i.e., $J \gg I$), and the role of the $I \times J$ matrix $\boldsymbol{M}$ is to project these rich high-dimensional features back down to the low-dimensional space containing the original features. The high-dimensional features $\boldsymbol{h}_t$ are obtained from $\boldsymbol{x}_t$ based on a universal background model (UBM) represented by a GMM, which we now describe in more details. We denote the concatenation of $\boldsymbol{x}_t$ with its $\Delta$ and $\Delta\Delta$ features, $\boldsymbol{x}_t^* \in \mathbb{R}^{3I}$, as

$$\boldsymbol{x}_t^* = \left[\boldsymbol{x}_t{}^\top, \Delta\boldsymbol{x}_t{}^\top, \Delta\Delta\boldsymbol{x}_t{}^\top\right]^\top. \tag{4.48}$$

A diagonal-covariance GMM for $\boldsymbol{x}_t^*$ is learned from the training data; the number of Gaussian components is denoted as $N_g$, and their mean and variance in dimension $i$ are denoted as $\mu_{n,i}$ and $\sigma_{n,i}$, respectively. Using this GMM, the high-dimensional features, $\boldsymbol{h}_t = \left[\boldsymbol{h}_{t,1}^\top, \ldots, \boldsymbol{h}_{t,N_g}^\top\right]^\top$, are computed from $\boldsymbol{x}_t^*$ as follows:

$$\boldsymbol{h}_{t,n} = p_{\mathrm{G}}(n|\boldsymbol{x}_t^*) \left[\frac{x_{t,1}^* - \mu_{n,1}}{\sigma_{n,1}}, \ldots, \frac{x_{t,3I}^* - \mu_{n,3I}}{\sigma_{n,3I}}, \xi\right]^\top, \tag{4.49}$$

where $p_{\mathrm{G}}(n|\boldsymbol{x}_t^*)$ is the posterior probability of the mixture component $n$ at frame $t$, and $\xi$ is a scaling factor for the bias term. Each sub-vector $\boldsymbol{h}_{t,n} \in \mathbb{R}^{3I+1}$ is a normalized and reweighted version of the feature vector based on the parameters and posterior of the $n$-th component. Although the number of total dimensions of feature $\boldsymbol{h}_t$ becomes very large in this setup, $\boldsymbol{h}_t$ is sparsified by setting to zero all but a given number of sub-vectors corresponding to the Gaussian components with the highest posterior probabilities $p_{\mathrm{G}}(n|\boldsymbol{x}_t^*)$.

The objective function with respect to the matrix $\boldsymbol{M}$ is obtained similarly to the previous sections by replacing $\boldsymbol{x}$ in the MMI and bMMI objective functions (Eqs. (4.37) and (4.38)) with the transformed feature $\boldsymbol{y}$, as follows:

$$\mathcal{F}_{\text{f-bMMI}}(\boldsymbol{M}) = \sum_r \ln \frac{p_\lambda\left(\boldsymbol{y}^{(r)}|\mathcal{H}_{s^{(r)}}\right)^\kappa p_L(s^{(r)})}{\sum_s p_\lambda\left(\boldsymbol{y}^{(r)}|\mathcal{H}_s\right)^\kappa p_L(s)\mathrm{e}^{-bA(s,s^{(r)})}}. \tag{4.50}$$

In our GMM systems, f-MMI/f-bMMI training with respect to $\boldsymbol{M}$ and MMI/b-MMI training with respect to the GMM parameter $\lambda$ are iteratively performed to optimize both parameters[1].

[1]Note that f-MMI/f-bMMI training is undertaken only for the GMM-based acoustic models, because the DNN acoustic models in Section 4.4.4 already include (non-linear) discriminative feature transformations in their deep networks.

Differentiating the objective function $\mathcal{F}$ by $\boldsymbol{M}$ as

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{M}} = \left[ \begin{array}{ccc} \frac{\partial \mathcal{F}}{\boldsymbol{y}_1} & \cdots & \frac{\partial \mathcal{F}}{\boldsymbol{y}_{T_f}} \end{array} \right] \left[ \begin{array}{ccc} \boldsymbol{h}_1 & \cdots & \boldsymbol{h}_{T_f} \end{array} \right]^\top, \tag{4.51}$$

where $T_f$ is the total number of frames of training data. The optimized matrix $\boldsymbol{M}$ is obtained by gradient descent using the (b)MMI statistics. Indirect differential of the objective function is given by

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{y}_t} = \sum_j \sum_m \frac{\gamma_{jm,t}^{ML}}{\sum_t \gamma_{jm,t}^{ML}} \left[ \frac{\partial \mathcal{F}}{\partial \boldsymbol{\mu}_{jm,t}} + 2 \frac{\partial \mathcal{F}}{\partial \boldsymbol{\Sigma}_{jm,t}} (\boldsymbol{y}_t - \boldsymbol{\mu}_{jm,t}) \right], \tag{4.52}$$

where $\gamma_{jm,t}^{ML}$ is the ML model posterior and $\frac{\partial \mathcal{F}}{\partial \boldsymbol{\mu}_{jm,t}}$ and $\frac{\partial \mathcal{F}}{\partial \boldsymbol{\Sigma}_{jm,t}}$ have been already obtained by the (b)MMI discriminative training of acoustic models [136]. To form the features, $N$ components of the GMM are obtained by clustering the Gaussians in the initial tri-phone acoustic models into $N$ components and re-estimating their parameters. The non-linear feature $\boldsymbol{h}_t$ [186] is calculated as

$$\boldsymbol{h}_{t,n} = \left[ p_{t,n} \frac{x_{t,1} - \mu_{n,1}}{\sigma_{n,1}}, \cdots, p_{t,n} \frac{x_{t,K} - \mu_{n,K}}{\sigma_{n,K}}, \beta p_{t,n} \right]^\top, \tag{4.53}$$

where $\mu_{n,k}$ and $\sigma_{n,k}$ are $k^{\text{th}}$ dimensional mean and standard deviation parameters of the $n^{\text{th}}$ Gaussian component. $\beta$ is the scaling factor. $p_{t,n}$ are Gaussian component posteriors computed for each frame, which are approximated such that all but the $N_1$-best posteriors are set to zero. This approximation is undertaken in order to reduce computational cost by ensuring that $\boldsymbol{h}_t$ is sparse.

## 4.5 Discriminative training methods of low-rank DNN

Although DNNs outperform GMMs, the number of parameters in DNNs tends to be greater than that in GMMs. For example, in the study of LVCSR task [153], for a GMM-based system, the number of HMM states is 3k and the mixture of Gaussian per state is 32; totally, the number of parameters is less than 10M. On the other hand, for DNN based system, the number of HMM states is the same, the number of nodes in each hidden layer is 2k, and the number of hidden layer is seven; totally, the number of parameters is over 30M. Thus the DNN model has three times larger number of parameters, which increases the computational cost and memory size.

There are some attempts to reduce a DNN model size [187, 188]. Xue *et al.* have proposed to apply SVD to DNN models and reduce the total number of parameters. Their method reduces the rank of the weight matrices and they show that SVD combined with fine-tuning is effective experimentally [188]. In their experiments, the speech data properties are not clear because their experiments were performed on private data. However typical LVCSR data uses close-talking microphones and so is relatively clean. Under reverberant and noisy environments in far-field conditions, DNN acoutic models need to be more complex to handle the increased variability of the signal. In this scenario, model reduction may have a negative effect on performance. Thus, the effectiveness of this technique on noisy reverberated speech needs to be evaluated.

Previous experiments on model reduction have focused on frame-level discriminative criteria such as CE. However, sequence-level discriminative training of acoustic models, using criteria such as MMI has improved the performance of conventional maximum likelihood based GMM models [146, 147, 175], as well as DNNs [189, 190, 191, 184, 192, 185, 193]. When combining the model reduction technique above with a sequence discriminative training, we need to investigate the effect of the order in which model reduction and sequence discriminative training are applied. For example it may be important to perform discriminative training after after model reduction in order to recover from loss of performance due to the approximation. We evaluate three approaches: the first approach is to apply SVD-based rank-reduction and fine-tuning for a CE full model and to perform discriminative training on a low-rank CE model; the second approach is to apply rank-reduction and fine-tuning for a MMI full model; the third approach is to perform discriminative training on the MMI low-rank model obtained from the second approach. This section investigates a several combinations of SVD reduction techniques with DNN sequence training experimentally for noisy reverberant speech recognition.

### 4.5.1 Reducing DNN model size singular value decomposition (SVD)

[188] proposed to use SVD to reduce the rank of the weight matrix $\boldsymbol{A}^l$ for a given layer $l$ to reduce the total number of parameters. Eq. (4.54) factorizes matrix $\boldsymbol{A}^l_{m\times n}$ as

$$\boldsymbol{A}^l_{m\times n} = \boldsymbol{U}_{m\times n}\boldsymbol{\Sigma}_{n\times n}\boldsymbol{V}^\top_{n\times n}. \tag{4.54}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix, whose elements are singular values arranged in a descending order ($\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$), $\boldsymbol{U}$ and $\boldsymbol{V}$ have orthonormal columns, and $\top$ denotes transpose. To reduce

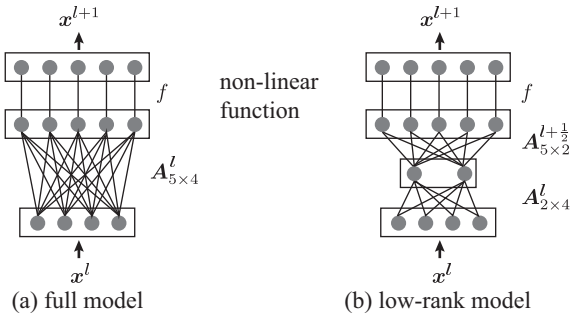



Fig. 4.4 Reducing DNN model parameters via low-rank factorization, from (a) $5 \times 4 = 20$ to (b) $5 \times 2 + 2 \times 4 = 18$.

the number of parameters of $\boldsymbol{A}_{m\times n}$, the $k$ largest singular values and their corresponding left and right singular vectors are used to from the low-rank factorization,

$$\begin{aligned}\boldsymbol{A}^l_{m\times n} &\approx \boldsymbol{U}_{m\times k}\boldsymbol{\Sigma}_{k\times k}\boldsymbol{V}^\top_{k\times n} \quad (k<n),\\ &= \left[\boldsymbol{U}_{m\times k}\sqrt{\boldsymbol{\Sigma}_{k\times k}}\right]\left[\sqrt{\boldsymbol{\Sigma}_{k\times k}}\boldsymbol{V}^\top_{k\times n}\right] = \boldsymbol{A}^{l+\frac{1}{2}}_{m\times k}\boldsymbol{A}^l_{k\times n}.\end{aligned} \tag{4.55}$$

Originally, computational costs of the matrix multiplication $\boldsymbol{Ax}$ are proportional to $O(mn)$. After low rank approximation, this becomes $O\left((m+n)k\right)$, so that computation is reduced for $k < mn/(m+n)$. The low rank approximation can be viewed as decomposing the $l$-th layer into two layers, the first a linear layer with weight matrix $\boldsymbol{A}^l_{k\times n}$, and the second a sigmoid layer with weight matrix, $\boldsymbol{A}^{l+\frac{1}{2}}_{m\times k}$, as shown in Fig. 4.4. In [188], $\boldsymbol{A}^l_{m\times n}$ is decomposed into the alternative factorization $\left[\boldsymbol{U}_{m\times k}\right]\left[\boldsymbol{\Sigma}_{k\times k}\boldsymbol{V}^\top_{k\times n}\right]$ which is functionally equivalent to (4.55). With offsets, the new layers become:

$$\begin{aligned}\boldsymbol{x}^{l+\frac{1}{2}} &= \boldsymbol{A}^l_{k\times n}\boldsymbol{x}^l + \boldsymbol{b}^l,\\ \boldsymbol{x}^{l+1} &= f\left(\boldsymbol{A}^{l+\frac{1}{2}}_{m\times k}\boldsymbol{x}^{l+\frac{1}{2}} + \boldsymbol{b}^{l+\frac{1}{2}}\right).\end{aligned} \tag{4.56}$$

where $\boldsymbol{b}^l$ is a $k$-dimensional vector initialized to zero, and $\boldsymbol{b}^{l+\frac{1}{2}}$ is the original $\boldsymbol{b}^l$. Fine tuning based on various discriminative objective functions can then be applied.

### 4.5.2 Combination of discriminative training with SVD

The order of discriminative training and model reduction is important and not trivial. Fig. 4.5 shows three approaches to generate discriminatively trained low-rank models, which we tested in this section. For all approaches, the initial model is a cross-entropy (CE) trained full model. The first approach, approach 1, is to apply SVD and fine-tuning for a CE full model and to perform discriminative training on a low-rank CE model; the second approach, approach 2, is to apply SVD and fine-tuning for a MMI full model; the third approach, approach 3, is to perform discriminative training on the MMI low-rank model obtained from approach 2.

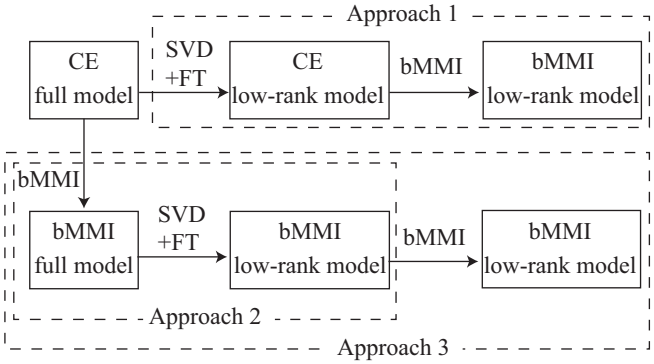



Fig. 4.5 Three approaches to generate MMI low-rank model with fine tuning (FT).

Table 4.6 DNN structure corresponding to SVD {1,2,3}.

| | input $\longrightarrow$ output |
|---|---|
| CE-full (2.85M) | $360 \times 331 + 331^2 \times 2 + 331 \times 8000$ |
| SVD1 (1.47M) | $360 \times \underline{100} + \underline{100} \times 331 + (331 \times \underline{96}) \times 2 \times 2$ $+ 331 \times \underline{162} + \underline{162} \times 8000$ |
| SVD2 (1.52M) | $360 \times 331 + (331 \times \underline{96}) \times 2 \times 2$ $+ 331 \times \underline{162} + \underline{162} \times 8000$ |
| SVD3 (1.59M) | $360 \times 331 + 331^2 \times 2 + 331 \times \underline{160} + \underline{160} \times 8000$ |
| SVD3 (1.91M) | $360 \times 331 + 331^2 \times 2 + 331 \times \underline{200} + \underline{200} \times 8000$ |

### 4.5.3 Experimental setups

We evaluated the performance on the second CHiME challenge Track 2 (details are in Section 5.2). In this case, we used noise-suppressed single-channel data obtained by prior-based binary masking (Section 5.2.2).

The settings of the acoustic features and feature transformation were as follows. We used the nnet2 of neural network training in the Kaldi toolkit [124]. The baseline features were 0th$\sim$12th order MFCCs + $\Delta$ + $\Delta\Delta$. Feature transformation techniques (LDA and MLLT) and speaker adaptation techniques in Section 4.2.3.2 were used to obtain 40-dimensional speaker-adapted features. The DNN input features were 9 consecutive frames of these feature concatenated into a 360-dimensional feature vector.

The number of the context-dependent HMM states was 1,989, which is equal to that of the last softmax layer outputs. The number of hidden layer was three. The initial learning rate for a CE full model was 0.01 and was decreased to 0.001 at the end of training. Starting from single-layer neural networks, we added layers one by one in every two iterations. One iteration used 400,000 samples. The total number of parameters was summarized in Table 4.6. In the CE training, the number of epoch was 15 for reducing learning rates and 5 for the constant final learning rate. Minibatch size was 128. After applying SVD to the CE full model or MMI full

Table 4.7 WER [%] on the CHiME challenge track 2 (si_dt_05) using DNN model showing the effectiveness of SVD and fine-tuning (FT) on noisy reverberated speech recognition. Initial model was CE-full model. Applying three types of SVD to this model, SVD {1,2,3} models were obtained. Input features were MFCC + LDA+MLLT + SAT+fMLLR (40 dimension × contiguous 9 frames).

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| CE-full (2.85M) | 53.44 | 42.40 | 34.53 | 27.94 | 24.77 | 20.49 | **33.93** |
| SVD1 (1.47M) wo FT | 58.95 | 48.52 | 40.15 | 34.33 | 31.05 | 26.10 | 39.85 |
| w FT | 53.81 | 42.88 | 35.58 | 28.74 | 25.36 | 21.92 | 34.72 |
| SVD2 (1.52M) wo FT | 59.06 | 48.59 | 40.12 | 34.30 | 31.07 | 26.08 | 39.87 |
| w FT | 52.68 | 42.06 | 34.34 | 28.29 | 24.96 | 20.58 | 33.82 |
| SVD3 (1.59M) wo FT | 58.93 | 48.12 | 39.98 | 33.97 | 30.48 | 25.36 | 39.47 |
| w FT | 51.82 | 41.04 | 32.64 | 26.42 | 23.63 | 19.87 | 32.57 |
| SVD3 (1.91M) wo FT | 57.09 | 46.72 | 38.91 | 33.04 | 29.08 | 24.07 | 38.15 |
| w FT | 51.76 | 40.67 | 32.87 | 26.21 | 23.79 | 19.86 | **32.53** |

model, fine-tuning needed 3 epochs for reducing learning rates from 0.001 to 0.0005 and 2 epochs for the constant final learning rate. For boosted MMI training, the learning rate was 0.001 when starting with the full CE model and 0.0001 for the low-rank models. The learning rate must be smaller for low-rank models than for full models because stochastic gradient descent tends to be less stable for low-rank models.

We evaluated three ways of applying SVD to full models: the first one was applying SVD to the all hidden layers (SVD 1); the second one was applying SVD to the all hidden layers except the first hidden layer because the first hidden layer has an important role for extracting features (SVD 2); the third one was applying SVD to the last layers, which have the largest number of parameters (SVD 3).

### 4.5.4 Results and discussion

#### 4.5.4.1 Which type of SVD is the best?

Table 4.7 shows the WER on si_dt_05. These models were all CE model without sequence discriminative training. After SVD, without FT, every low-rank model degraded significantly. Fine-tuning greatly improved the performance of all models, consitent with the results of [188]. Among them, the SVD3 type of decomposition was the best. The performance of SVD1 was inferior to that of SVD2. This indicates that the weight matrices in the first layer had higher effective rank than those in the upper layers.

#### 4.5.4.2 Which type of discriminative training approach is the best?

Table 4.8 shows the results of discriminatively trained models. Sequence discriminative training led to significant improvements for the full model. In approach 1, The performance improvement

Table 4.8 WER [%] on the CHiME challenge track 2 (si_dt_05) using DNN model showing the effectiveness of sequence discriminative training. Initial model is Cross-entropy (CE) model. Three types of approaches were evaluated.

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| bMMI-full (2.85M) | 48.37 | 36.66 | 30.15 | 24.18 | 20.71 | 17.27 | **29.56** |
| *Approach 1 (from CE low-rank model) | | | | | | | |
| SVD1 (1.47M) bMMI | 47.87 | 37.62 | 30.61 | 24.43 | 21.23 | 18.07 | 29.97 |
| SVD2 (1.52M) bMMI | 47.38 | 36.47 | 29.30 | 24.00 | 20.64 | 17.32 | 29.19 |
| SVD3 (1.59M) bMMI | 46.36 | 35.11 | 28.06 | 23.03 | 19.41 | 16.48 | **28.08** |
| SVD3 (1.91M) bMMI | 47.03 | 35.31 | 28.38 | 22.82 | 19.53 | 16.77 | 28.31 |
| *Approach 2 (from bMMI full model) | | | | | | | |
| SVD1 (1.47M) wo FT | 54.61 | 43.30 | 35.79 | 30.80 | 27.25 | 22.39 | 35.69 |
| w FT | 53.25 | 42.51 | 34.93 | 28.81 | 25.30 | 21.71 | 34.42 |
| SVD2 (1.52M) wo FT | 54.82 | 43.27 | 35.89 | 30.83 | 27.28 | 22.54 | 35.77 |
| w FT | 52.80 | 41.64 | 34.39 | 27.70 | 24.56 | 20.96 | 33.68 |
| SVD3 (1.59M) wo FT | 54.08 | 42.13 | 34.64 | 29.41 | 25.87 | 21.79 | 34.65 |
| w FT | 51.67 | 41.26 | 33.15 | 26.60 | 23.57 | 19.66 | 32.65 |
| SVD3 (1.91M) wo FT | 52.97 | 41.07 | 33.94 | 27.66 | 24.72 | 20.77 | 33.52 |
| w FT | 51.60 | 40.64 | 33.13 | 26.61 | 23.51 | 19.72 | 32.54 |
| *Approach 3 (from Approach 2 model) | | | | | | | |
| SVD1 (1.47M) bMMI | 48.61 | 37.81 | 30.82 | 25.20 | 21.52 | 18.47 | 30.41 |
| SVD2 (1.52M) bMMI | 48.10 | 36.95 | 30.54 | 23.82 | 21.20 | 17.54 | 29.69 |
| SVD3 (1.59M) bMMI | 47.71 | 36.91 | 29.36 | 23.35 | 20.56 | 16.96 | 29.14 |
| SVD3 (1.91M) bMMI | 47.74 | 37.14 | 29.34 | 23.31 | 20.55 | 17.14 | 29.20 |

of low-rank CE model was larger than CE full model, which is reported in general discriminative training studies for speech recognition that smaller models have bigger improvement [194]. In approach 2, without FT, the performance of bMMI low-rank model was better than that of CE low-rank model without FT, however, for the bMMI low-rank model, FT was less effective. In approach 3, discriminative training on the bMMI low-rank model again improved the performance but was less effective than for CE low-rank model perhaps due to over-training. Overall approach 1 was the best.

#### 4.5.4.3 Evaluation set

Table 4.9 shows the results on evaluation set (si_et_05). Tendencies were the same to the development set. SVD 3 types of decomposition was effective and their performance was superior to that of the original bMMI full model by 1% absolute.

### 4.5.5 Conclusion

To reduce the number of DNN parameters, a model reduction technique using low-rank approximation has been applied to noisy reverberant speech recognition. Experiments demonstrate that low-rank approximation of the last layer of DNN or all layers except the first layer is more effective than rank reduction of all layers. Sequence discriminative training further improved

Table 4.9 WER [%] on the CHiME challenge track 2 (si_et_05) using DNN model showing the effectiveness of sequence discriminative training. Initial model is cross-entropy (CE) model.

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| CE-full (2.85M) | 44.48 | 35.72 | 29.46 | 21.99 | 16.63 | 15.34 | 27.27 |
| bMMI-full | 39.02 | 28.94 | 23.39 | 18.27 | 13.94 | 11.96 | **22.59** |
| *Approach 1 (from CE low-rank model) | | | | | | | |
| SVD1 (1.47M) bMMI | 40.03 | 29.67 | 23.97 | 18.51 | 14.40 | 12.74 | 23.22 |
| SVD2 (1.52M) bMMI | 39.47 | 28.41 | 23.11 | 18.16 | 13.53 | 12.11 | 22.47 |
| SVD3 (1.59M) bMMI | 37.94 | 27.59 | 22.53 | 17.39 | 12.87 | 10.97 | **21.55** |
| SVD3 (1.91M) bMMI | 37.51 | 27.65 | 22.42 | 17.52 | 12.82 | 11.47 | 21.57 |

performance. The most effective combination of discriminative training with model reduction was to reduce the base model first and then to perform discriminative training on the low-rank model. This discriminatively trained low-rank model outperformed the discriminatively trained full model.

## 4.6 Discriminative training methods of system combination

Many researchers have pointed out that combining different systems effectively improves performance (e.g., Recognizer Output Voting Error Reduction (ROVER) [195] and [196, 197, 198]) even if the performance of the complementary systems is lower than that of the base system. Because effective system combination relies on a combination of hypotheses with different trends, generally, different features or training methods are used to construct complementary systems [199, 200, 201, 202]. For example, the random forest approach [199] is a simple way of constructing complementary systems, which builds multiple shared tri-phone trees by randomly changing the topologies of existing trees. Especially for Deep Neural Networks (DNN), to avoid local minimum problems, random initialization and averaging of multiple model parameters are generally used to improve the performance of the original single system. However, system combinations do not necessarily improve the performance when the hypotheses of complementary systems have similar trends or yield too many errors (as we also confirmed in our experiments). Classical system combination approaches require trial-and-error attempts because they do not rely on a general theoretical background such as an objective function in discriminative training [203, 175, 204].

To address this problem, complementary system training algorithm of acoustic models for system combination based on the MPE criterion has been proposed [205]. This lattice-based approach provides theoretical background for training complementary systems and is promising because conventional discriminative training methods can be easily applied. We also proposed a method to discriminatively train acoustic models based on the MMI criterion in order to clarify the relationship between the reference and hypotheses of the base and complementary system further [206].

In this section, we extend the above approach and propose a general framework of sequential discriminative training for system combination encompassing various model training methods such as acoustic modeling, here applied to GMM and DNN, as well as discriminative feature transformation. Our method generalizes the objective function of discriminative training in order to balance the objective function given by correct labels and that given by the hypotheses of the base systems. The advantages of our proposed method are the fact it leads to a simple extension of conventional lattice-based discriminative training and its clear resemblance to a discriminative training method. In addition, because the formulation of our proposed method includes margin-based discriminative training, one can adjust the degree of deviation of the complementary systems' outputs with respect to those of the base systems. Thus, the effectiveness of the proposed approach covers the wide area of discriminative acoustic modeling and feature transformation.

Section 4.6.1 first describes the general discriminative training framework for complementary systems. Then, we apply this framework to sequential discriminative training of acoustic models (GMM and DNN) and discriminative feature transformation in Sections 4.4.2 and 4.4.6, respectively. Experiment in Section 4.6.4 shows the effectiveness of the proposed approach ex-

perimentally.

### 4.6.1 Generalized discriminative training framework for complementary systems

In this section, complementary systems are constructed by discriminatively training a model starting from an initial model. The proposed discriminative training method for complementary systems is extended from a discriminative training principle. Assuming $Q$ base systems have already been constructed, the discriminative training objective function $\mathcal{F}$ is generalized to the following proposed objective function $\mathcal{F}^c$, which subtracts from the original objective function involving the correct labels $\omega_r$, the objective functions involving the 1-best hypotheses (lattice) $\omega_{q,1}$ $(q = 1, \ldots, Q)$ of the $Q$ base systems:

$$\mathcal{F}^c_\varphi(\omega_r, \omega_{q,1}) = (1+\alpha)\mathcal{F}_\varphi(\omega_r) - \frac{\alpha}{Q}\sum_{q=1}^{Q}\mathcal{F}_\varphi(\omega_{q,1}), \tag{4.57}$$

where $\varphi$ is the set of model parameters of a complementary system to be optimized and $\alpha$ is a scaling factor. The 1-best hypotheses can be easily obtained by the lattice rescoring. If $\alpha$ equals zero, this objective function matches that of classical discriminative training. The first term in Eq. (4.57) promotes good performance according to the discriminative training criterion, whereas the second term makes the target system generate hypotheses that have a different tendency from the original base models. The next sections provide concrete forms of the objective function and model parameters in Eq. (4.57) for acoustic modeling problems and discriminative feature transformation.

### 4.6.2 Complementary acoustic model training

This section applies the MMI criterion to the above-mentioned framework. MMI training aims to maximize the objective function 4.37. For simplicity, the number of base systems $Q$ is taken as one below, and the index $q$ is omitted. In the MMI criterion, we replace $\varphi$ by $\lambda_c$ and $\mathcal{F}$ by $\mathcal{F}^{\mathrm{MMI}}$ in Eq. (4.57) to obtain:

$$\mathcal{F}^{\mathrm{c}}_{\lambda_c}(\omega_r, \omega_1) = \mathcal{F}^{\mathrm{MMI}}_{\lambda_{\mathrm{c}}}(\omega_r) + \alpha \ln \frac{P_{\lambda_c}(\omega_r, \boldsymbol{X})}{P_{\lambda_c}(\omega_1, \boldsymbol{X})}, \tag{4.58}$$

which is a new objective function for a complementary system within an MMI discriminative training framework, that has an additional log-likelihood ratio term.

In boosted MMI (bMMI) [175], the standard MMI objective function is shown in Eq. 4.38. As a simple extension of the Eq. (4.58), by replacing $\mathcal{F}^{\mathrm{MMI}}$ with $\mathcal{F}^{\mathrm{bMMI}}$, and adding the (reverse sign) boosting factors to the log-likelihood ratio term analogous to Eq. (4.38), we can introduce

the following objective function [2] [3]:

$$\mathcal{F}^{\mathrm{c}}_{\lambda_c}(\omega_r,\omega_1)=\mathcal{F}^{\mathrm{bMMI}}_{\lambda_{\mathrm{c}}}(\omega_r)+\alpha\ln\frac{\sum_{s_r\in\mathcal{S}_{\omega_r}}p_\lambda\left(s_r,\boldsymbol{X}\right)^\kappa p_L(\omega_r)}{\sum_{s_1\in\mathcal{S}_{\omega_1}}p_\lambda\left(s_1,\boldsymbol{X}\right)^\kappa p_L(\omega_1)e^{b_1A(s_1,s_r)}},\tag{4.59}$$

where $s_1$ is an HMM state sequence corresponding to the 1-best hypothesis of the base system $\omega_1$. The reverse sign boosting factor $b_1$ is discussed in the Section 4.6.2.1. This procedure is commonly used to obtain the objective functions of acoustic modeling (GMM and DNN) and discriminative feature transformation in this section.

#### 4.6.2.1 GMM

We now explain the update equation of the complementary system by using the proposed objective function (4.59). The update formulae for the mean and covariance of GMM take the same form as the original (b)MMI formulae (4.39) up to simply modifying the variables as ($\gamma^{num}_{jm,t}$ is unchanged)

$$\begin{aligned}\Delta'_{jm,t}&=(1+\alpha)\left(\gamma^{num}_{jm,t}-\gamma^{den\ \prime}_{jm,t}\right),\\ \gamma^{den\ \prime}_{jm,t}&=\frac{\gamma^{den}_{jm,t}+\alpha\gamma^{1}_{jm,t}}{1+\alpha},\\ D'_{jm}&=\frac{D_{jm}}{1+\alpha}.\end{aligned}\tag{4.60}$$

To elucidate the effect of the $b_1$ term, we first consider for simplicity, the single-frame classification problem of the proposed approach, which is approximated by assuming an utterance has only one frame. In a single frame, because we do not need to consider the HMM states transition, posteriors are proportional to the product of acoustic and language scores multiplied by the boosting factors. In this case, the index $t$ is omitted, and $\gamma^1_{jm}$ can be represented by

$$\gamma^1_{jm}=\begin{cases}C^1_{jm}e^{b_1} & (j,\ s.t.,\ s_1=s_r,\ \text{correct}),\\ C^1_{jm} & (j,\ s.t.,\ s_1\neq s_r,\ \text{incorrect}),\end{cases}\tag{4.61}$$

$$\begin{aligned}C^1_{jm}&=\frac{p_\lambda\left(j,m,\boldsymbol{x}\right)^\kappa p_L(\omega_1)}{\sum_{m',j'_1\in\mathcal{S}_{\omega_1}}p_\lambda\left(j'_1,m',\boldsymbol{x}\right)^\kappa p_L(\omega_1)e^{b_1A(j'_1,j_r)}},\\ p_\lambda\left(j,m,\boldsymbol{x}\right)&=\pi_{jm}\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{jm},\boldsymbol{\Sigma}_{jm}),\end{aligned}\tag{4.62}$$

where $\mathcal{N}$ is a probability density of a single Gaussian and $j_1$ and $j_r$ are the HMM states obtained from the 1-best hypotheses of the base system and the label, respectively. The factor $b_1$ decreases $\gamma^1_{jm}$ in the case that the base system gives incorrect hypotheses. Because $\gamma^1_{jm}$ is subtracted in Eq. (4.60), diminishing it increases $\Delta_{jm,t}$ for these hypotheses. This is analogous to boosting

[2] There is another derivation obtained by substituting the bMMI criterion Eq. (4.38) into our generalized form Eq. (4.57). We will further investigate the relationship in our future work.

[3] Note that because there are multiple HMM state sequences realizing the same phoneme/word sequence, the denominator of the second term in Eq. (4.59) is obtained by the summation over these multiple sequences, and thus the boosting factor $b_1$ do affect the optimization.

algorithms such as AdaBoost [207, 208], which assign larger weights to data points where the base system gives incorrect hypotheses.

For the sequential case, it is difficult to show a direct relationship between the posterior and the boosting factors as in the single frame case because, to gather posteriors, the forward-backward algorithm is used and posteriors at the current frame are affected by the previous and future frames. However, similarly to the discussion in the single frame case, because the posterior $\gamma^1_{jm,t}$ is an increasing function of the base system's sentence average accuracy, even in the sequential case, the proposed method has a relationship to boosting.

Algorithm 1 shows the proposed algorithm for updating a complementary system model by using the extended Baum-Welch (EBW) algorithm or gradient descent (GD). In this section, EBW algorithm was used.

**Algorithm 1** Construct complementary system model for GMM

**Input:** Initial model $\lambda$ (e.g., ML), base system models $\lambda_q$, numerator ($\omega_r$ aligned) lattice $\mathcal{A}$, and denominator lattice $\mathcal{L}$ of Eq. (4.37) or (4.38)

**for** $i = 1$ to $i_{eb}$ **do**

Rescore $\mathcal{A}$ and $\mathcal{L}$ with $\lambda$

$\gamma^{num}_{jm,t}$ and $\gamma^{den}_{jm,t} \Leftarrow$posteriors are gathered on $\mathcal{A}$ and $\mathcal{L}$, respectively

$\gamma_{jm,t} \Leftarrow -\gamma^{den}_{jm,t} + (1+\alpha)\gamma^{num}_{jm,t}$

**for** $q = 1$ to $Q$ **do**

Rescore $\mathcal{L}$ with $\lambda_q$

$\mathcal{L}_1 \Leftarrow$best path of $\mathcal{L}$

Rescore $\mathcal{L}_1$ with $\lambda$

$\gamma^1_{jm,t} \Leftarrow$ posteriors are gathered on $\mathcal{L}_1$

$\gamma_{jm,t} \Leftarrow -\frac{\alpha}{Q}\gamma^1_{jm,t} + \gamma_{jm,t}$

**end for**

$\gamma^{num}_{jm,t}, \gamma^{den}_{jm,t} \Leftarrow$positive and negative parts of $\gamma_{jm,t}$

$\lambda \Leftarrow$ Update $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by EBW or GD (Eq. (4.39))

**end for**

**Output:** Complementary system model ($\lambda_c \leftarrow \lambda$)

#### 4.6.2.2 DNN

For the proposed method, the denominator posterior is modified by Eq. (4.60) as in the GMM case. The gradients for all the DNN parameters are derived from Eq. (4.44) based on the back-propagation procedure.

Algorithm 2 shows that the method for constructing complementary system models for DNN is similar to the GMM case. This versatility is one of the advantages of the proposed generalized framework.

**Algorithm 2** Construct complementary system model for DNN

**Input:** Initial model $\theta$, base system models $\theta_q$, numerator ($\omega_r$ aligned) lattice $\mathcal{A}$, and denominator lattice $\mathcal{L}$ of Eq. (4.37) or (4.38)

**for** $i = 1$ to $i_{eb}$ **do**

Rescore $\mathcal{A}$ and $\mathcal{L}$ with $\theta$

$\gamma_{j,t}^{num}$ and $\gamma_{j,t}^{den} \Leftarrow$posteriors are gathered on $\mathcal{A}$ and $\mathcal{L}$, respectively

$\gamma_{j,t} \Leftarrow -\gamma_{j,t}^{den} + (1+\alpha)\gamma_{j,t}^{num}$

**for** $q = 1$ to $Q$ **do**

Rescore $\mathcal{L}$ with $\theta_q$

$\mathcal{L}_1 \Leftarrow$best path of $\mathcal{L}$

Rescore $\mathcal{L}_1$ with $\theta$

$\gamma_{j,t}^{1} \Leftarrow$ posteriors are gathered on $\mathcal{L}_1$

$\gamma_{j,t} \Leftarrow -\frac{\alpha}{Q}\gamma_{j,t}^{1} + \gamma_{j,t}$

**end for**

$\gamma_{j,t}^{num}, \gamma_{j,t}^{den} \Leftarrow$positive and negative parts of $\gamma_{j,t}$

$\theta \Leftarrow$ Update $a$ by EBW or GD (Eq. (4.44))

**end for**

**Output:** Complementary system model ($\theta_c \leftarrow \theta$)

### 4.6.3 Complementary discriminative feature transformation

This framework can be applied for discriminative feature transformation (Section 4.4.6). As in the GMM case, the objective function for complementary systems is introduced from Eq. (4.57) by replacing $\varphi$ by $\boldsymbol{M}_c$ and $\mathcal{F}$ by $\mathcal{F}^{\text{f-MMI}}$ ($b = 0$ for Eq. (4.50)) as

$$\mathcal{F}^{\text{c}}_{\boldsymbol{M}_c}(\omega_r, \omega_1) = \mathcal{F}^{\text{f-MMI}}_{\boldsymbol{M}_c}(\omega_r) + \alpha \ln \frac{P_{\boldsymbol{M}_c}(\omega_r, \boldsymbol{Y})}{P_{\boldsymbol{M}_c}(\omega_1, \boldsymbol{Y})}, \tag{4.63}$$

and, in the same procedure from Eq. (4.58) to Eq. (4.59), the boosted version of Eq. (4.63) is given by

$$\mathcal{F}^{\text{c}}_{\boldsymbol{M}_c}(\omega_r, \omega_1) = \mathcal{F}^{\text{f-bMMI}}_{\boldsymbol{M}_c}(\omega_r) + \alpha \ln \frac{\sum_{s_r \in \mathcal{S}_{\omega_r}} p_{\boldsymbol{M}_c}(s_r, \boldsymbol{Y})^{\kappa} p_L(\omega_r)}{\sum_{s_1 \in \mathcal{S}_{\omega_1}} p_{\boldsymbol{M}_c}(s_1, \boldsymbol{Y})^{\kappa} p_L(\omega_1) e^{-b_1 A(s_1, s_r)}}. \tag{4.64}$$

Thus the proposed framework can be applied to the discriminative feature transformation for a complementary system starting from the generalized objective function.

Algorithm 3 shows the proposed algorithm for updating a complementary system model by using the gradient descent algorithm.

### 4.6.4 Experimental setups

We evaluated the performance improvement provided by these system combination techniques on two corpus: the second CHiME challenge Track 2 (details are in Section 5.2) and CSJ (Section 4.3.3). The former aimed to validate the performance of the proposed method for acoustic

**Algorithm 3** Construct complementary system model for f-MMI

**Input:** Acoustic model $\lambda$, initial matrix $\boldsymbol{M}$, base system matrix $\boldsymbol{M}_q$, numerator ($\omega_r$ aligned) lattice $\mathcal{A}$, and denominator lattice $\mathcal{L}$ of Eq. (4.50)

**for** $i = 1$ to $i_{eb}$ **do**

Rescore $\mathcal{A}$ and $\mathcal{L}$ with $\lambda$ using $\boldsymbol{y}_t$ $(= \boldsymbol{x}_t + \boldsymbol{M}\boldsymbol{h}_t)$

$\gamma_{jm,t}^{num}$ and $\gamma_{jm,t}^{den} \Leftarrow$posteriors of $\mathcal{A}$ and $\mathcal{L}$, respectively

$\gamma_{jm,t} \Leftarrow -\gamma_{jm,t}^{den} + (1+\alpha)\gamma_{jm,t}^{num}$

**for** $q = 1$ to $Q$ **do**

Rescore $\mathcal{L}$ with $\lambda$ using $\boldsymbol{y}_t$ $(= \boldsymbol{x}_t + \boldsymbol{M}_q\boldsymbol{h}_t)$

$\mathcal{L}_1 \Leftarrow$best path of $\mathcal{L}$

Rescore $\mathcal{L}_1$ with $\lambda$

$\gamma_{jm,t}^{1} \Leftarrow$ posterior of $\mathcal{L}_1$

$\gamma_{jm,t} \Leftarrow -\frac{\alpha}{Q}\gamma_{jm,t}^{1} + \gamma_{jm,t}$

**end for**

$\gamma_{jm,t}^{num}, \gamma_{jm,t}^{den} \Leftarrow$positive and negative parts of $\gamma_{jm,t}$

$\boldsymbol{M} \Leftarrow$ Update elements in $\boldsymbol{M}$ by calculating the indirect differential in Eq. (4.52)

**end for**

**Output:** Complementary system matrix ($\boldsymbol{M}_c \leftarrow \boldsymbol{M}$)

modeling (GMM and DNN) and discriminative feature transformation and the effectiveness of our proposed generalized framework experimentally. In this case, we used noise-suppressed single-channel data obtained by prior-based binary masking (Section 5.2.2). The latter aimed to show that the proposed method is effective for other tasks and the performance improvement is independent on tasks.

The baseline features were both MFCC and perceptual linear prediction (PLP) (0-12 order MFCCs/PLPs + $\Delta$ + $\Delta\Delta$). Feature transformation techniques (LDA and MLLT) and speaker adaptation technique in Section 4.2.3.2 were used. The number of the context-dependent HMM states was 2,500 and the total number of Gaussians was 15,000. Tree structures were different between MFCC and PLP features, the latter of which also considered a random forest-like effect. For the DNN, we used the nnet2 of neural network training implemented in Kaldi with 3 hidden layers and 1,000,000 parameters. The initial learning rate was 0.01 and was decreased to 0.001 at the end of training. In discriminative feature transformation, 400 Gaussians were used and offset features were calculated for each of the 40 dimensional features with context expansion (9 frames). The dimension of the feature vector $\boldsymbol{h}_t$ was $400 \times 40 \times 9$. Features with the top 2 posteriors were selected and all other features were ignored. $\beta$ was set to 5. For the proposed method, parameters $\alpha$ and $b_1$ were 0.75 and 0.3, which were optimized by using the development set.

For CSJ task, the ASR settings were similar to the CHiME challenge, but the language model size was about 70k and the number of the context-dependent HMM states was 3,500 and the total number of Gaussians was 96,000. Test set 1 contained about 10-15 minutes lecture by 10

Table 4.10 Average WER[%] for isolated speech (**si_dt_05** and **si_et_05**) on acoustic modeling (GMM). (MFCC and PLP with LDA+MLLT+SAT+fMLLR) (upper: conventional Single systems (S), upper middle: ROVER among conventional multiple systems (R), lower middle: single Proposed complimentary systems (P), and lower: ROVER including Proposed complementary system (RP))

| ID | MFCC | | | PLP | | | WER | |
|---|---|---|---|---|---|---|---|---|
| | ML | bMMI | $\text{bMMI}_\text{c}$ | ML | bMMI | $\text{bMMI}_\text{c}$ | (dt) | (et) |
| S1 | ✓ | | | | | | 38.15 | 32.20 |
| S2 | | ✓ | | | | | 35.86 | 29.46 |
| S3 | | | | ✓ | | | 38.10 | 32.23 |
| S4 | | | | | ✓ | | 36.43 | 29.98 |
| R1 | ✓ | ✓ | | | | | 36.06 | 29.26 |
| R2 | ✓ | ✓ | | ✓ | ✓ | | 34.97 | 28.00 |
| P1 | | | ✓ | | | | 36.21 | 30.09 |
| P2 | | | | | | ✓ | 36.72 | 30.46 |
| RP1 | | ✓ | ✓ | | | | 35.67 | 28.80 |
| RP2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 34.55 | 27.49 |

Table 4.11 Average WER[%] for isolated speech (**si_dt_05**, **si_et_05**) on discriminative feature transformation. (MFCC with LDA+MLLT and SAT+fMLLR)

| ID | bMMI | f-bMMI | f-$\text{bMMI}_\text{c}$ + $\text{bMMI}_\text{c}$ | f-$\text{bMMI}_\text{c}$ +bMMI | WER | |
|---|---|---|---|---|---|---|
| | | | | | (dt) | (et) |
| S5 | ✓ | | | | 35.86 | 29.46 |
| S6 | | ✓ | | | **33.19** | **27.00** |
| R3 | ✓ | ✓ | | | 33.80 | 27.15 |
| P3 | | | ✓ | | 35.38 | 28.27 |
| P4 | | | | ✓ | 33.88 | 27.86 |
| RP3 | | ✓ | ✓ | | 32.75 | **26.60** |
| RP4 | | ✓ | | ✓ | **32.67** | 26.62 |

different male speakers. The parameters for the proposed method are the same to those for the CHiME challenge.

We used ROVER for combining output hypotheses from multiple systems. Certainly, especially for two systems, confusion network combination (CNC) is better than ROVER, however, ROVER is more simple and can be applied for many systems.

## 4.6.5 Results and discussion

### 4.6.5.1 CHiME challenge (Noise robust ASR)

For the GMM system, although detailed results are shown in [206], we briefly describe the results for comparison with the other approaches. Table 4.10 shows the WER using MFCC and PLP features with the feature transformation of LDA+MLLT and SAT+fMLLR. The upper,

Table 4.12 Average WER[%] for isolated speech (**si_dt_05**, **si_et_05**) on acoustic modeling (DNN). (MFCC with LDA+MLLT)

| ID | DNN | bMMI | $\text{bMMI}_\text{c}$ | WER (dt) | WER (et) |
|---|---|---|---|---|---|
| S7 | ✓ | | | 36.59 | 30.84 |
| S8 | | ✓ | | **32.40** | **26.91** |
| P5 | | | ✓ | 33.09 | 27.97 |
| RP5 | | ✓ | ✓ | **31.38** | **26.48** |

Table 4.13 WER[%] in terms of SNR[dB] for isolated speech (**si_et_05**) on f-bMMI (S6→RP3) and DNN (S8→RP5).

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| S6 | 44.14 | 35.42 | 28.56 | 21.46 | 17.41 | **14.98** | 27.00 |
| S8 | 43.86 | 33.36 | 28.13 | 22.01 | 17.75 | 16.36 | 26.91 |
| RP3 | 43.21 | 34.24 | 28.25 | 21.58 | **17.17** | 15.13 | 26.60 |
| RP5 | **42.85** | **32.43** | **27.91** | **21.56** | 17.75 | 16.40 | **26.48** |

upper middle, lower middle, and lower sections correspond to conventional single systems (S1-S4), ROVER among conventional multiple systems (R1,R2), proposed complementary systems (P1,P2), and ROVER including proposed complementary systems (RP1,RP2), respectively. The performances of proposed complementary systems (P1,P2) were in between that of ML(S1,S3) and that of bMMI(S2,S4). Because the performance of ML was much lower than that of bMMI, the combination with the ML model was not effective for ROVER (S2→R1). In this case, even though the numbers of systems were the same (two) for both cases, the performance of the combination of bMMI and $\text{bMMI}_\text{c}$ (RP1) was higher than that of the combination of ML and bMMI (R1) because the performance of $\text{bMMI}_\text{c}$ was moderate, which made the system combination effective. This is an advantage of the performance adjustability of the proposed method. Adding two systems to the conventional ROVER using four systems further improved the WER by 0.42%(dt) and 0.51%(et) (R2→RP2). Because the hypotheses of MFCC systems are quite different from those of PLP systems, alternative update of the complementary system for both feature systems could not improve the performance.

In addition, we validated the discriminative feature transformation and DNN on the development set. Table 4.11 (left column) shows the WER using discriminative feature space transformation on top of MFCC features with the feature transformation of LDA+MLLT and SAT+fMLLR. f-bMMI is usually combined with discriminative training of GMM (i.e., bMMI). In this case, we constructed complementary systems in two ways: for both f-bMMI and bMMI, the objective functions were modified (i.e., f-$\text{bMMI}_\text{c}$ + $\text{bMMI}_\text{c}$ using Eqs. (4.64) and (4.59)) or only for f-bMMI, the objective function was modified (i.e., f-$\text{bMMI}_\text{c}$ + bMMI using Eqs. (4.64) and (4.38)). The performance of the combination of bMMI and f-bMMI (R3) was lower than that of f-bMMI only, but the combination with the proposed complementary systems (RP3 and RP4) improved the accuracy. There was no significant difference between f-$\text{bMMI}_\text{c}$ + bMMI

Table 4.14 Average WER[%] (CSJ, test set 1) on acoustic modeling (GMM). (MFCC)

| ID | ML | bMMI | $bMMI_c$ | WER |
|---|---|---|---|---|
| S1 | ✓ | | | 21.00 |
| S2 | | ✓ | | **18.64** |
| R1 | ✓ | ✓ | | 18.69 |
| P1 | | | ✓ | 18.81 |
| RP1 | | ✓ | ✓ | 18.52 |
| RP2 | ✓ | ✓ | ✓ | **18.28** |

and f-$bMMI_c$ + $bMMI_c$. Table 4.12 shows the WER using DNN on top of MFCC and PLP features with the feature transformation of LDA+MLLT. Discriminative training improved the accuracy by 4.19% (S7→S8) significantly. Combination with the proposed method also improved the accuracy further (RP5).

We also validated the performance on the evaluation set, and confirmed the similar experimental tendencies. Table 4.13 further investigates the WER in terms of SNR by comparing S6 with RP3 (f-bMMI case) and S8 with RP5 (DNN case). For almost all the cases, the proposed method improved the WER, especially for the low SNR cases (1.2% maximum). Thus, the performance improvements were stable and robust in different environments.

In conclusion, the experimental results confirmed the effectiveness of the proposed approach for a wide range of sequential discriminative training methods for acoustic modeling and feature transformation.

#### 4.6.5.2 CSJ (LVCSR)

The performance was evaluated on a second corpus (CSJ). This task did not include noises but it was composed of spontaneous speech and the vocabulary size was much larger than the CHiME challenge (WSJ0). Table 4.14 shows the WER for the test set 1 by using the proposed GMM training. In this case, conventional ROVER (R1) decreased the performance from the single system (S2), however, the proposed method using two or three systems improved the accuracy by 0.36%. Thus, the proposed approach was also effective for large-scale spontaneous speech recognition.

### 4.6.6 Conclusion

We proposed a general discriminative training framework for system combination. The proposed method can construct complementary systems in the framework of discriminative training methods, and it is capable of improving the WER on reverberated and highly noisy speech as well as large vocabulary spontaneous speech recognition tasks. Moreover, it is effective for discriminative training of acoustic models (GMM and DNN) and discriminative feature transformation. In future work, the proposed method will be combined with other discriminative techniques, such as acoustic modeling with other discriminative criteria and discriminative language modeling [203].

## 4.7 Discriminative training methods of language models

Neural networks have been recently introduced and used for language processing. Among them, the recurrent neural network based language model (RNN-LM) has become popular due to its high performance [209, 210] as well as the availability of open source software [211, 212]. RNN is a neural network (NN) that contains one or more hidden layers with recursive inputs. Although their computational costs are high, RNN-LM greatly improves ASR performance. The greatest difference between RNN-LM and conventional $n$-gram models is the available word context length [213]. The role of a language model is to estimate posterior probabilities of target words based on previous words context. A long context provides much information. However, the simple use of a long context (i.e., 4-gram or 5-gram) by a conventional $n$-gram language model encounters data sparsity problems. To address these problems, RNN-LM first maps a high-dimensional 1-of-N representation of a target word to a low-dimensional continuous space in a hidden layer and directly estimates the posterior probability of the target word. The hidden-layer units from the previous frame are then connected to the input vector in the next frame. These recursive inputs collect the history of words in the low-dimensional hidden-layer units. RNN-LM implicitly considers the entire history of words, whereas widely used $n$-gram models consider only previous $(n-1)$ words. Although there are several trials [214, 215, 216], using RNN-LM directly for decoding is essentially difficult because feed-forward propagation of RNN is much more expensive than using a table lookup method with an n-gram model [217]. Therefore, RNN-LM is typically used for post-processing such as N-best or lattice rescoring.

However, the training criteria of RNN-LM are based on CE between predicted and reference words. That is, the CE criterion does not explicitly consider discriminative criteria calculated from ASR hypotheses and references. On the other hand, discriminative criteria show the effectiveness in GMM-based acoustic model and feature transformation training at accomplishing various ASR tasks [175, 203, 218, 219]. Moreover, those for DNN acoustic modeling can also reduce ASR errors, while maintaining a fundamental high frame-level discriminability [191, 184, 192, 185, 193]. RNN-LM CE criterion is discriminative in the sense of considering the posterior distribution of a target word given history, but a discriminative criterion of RNN-LM that considers ASR hypotheses can further correct ASR errors. In recent years, [220] and [221] have applied sequence discriminative training to RNN acoustic modeling and natural language understanding, respectively. In this study, we propose a new discriminative training method for RNN-LM[4].

Another discriminative model within N-best rescoring framework is known as a discriminative language modeling (DLM) [204, 226, 227]. DLM is a corrective training method based on n-gram counts obtained from reference and ASR hypothesis examples of training data. It can correct errors that are inherent to a decoder in an efficient manner especially for words of short context. However, the context of DLM is limited to an $n$-gram (usually a tri-gram) that is identical to that in a typical $n$-gram language model. In addition, a long context cannot be used for error

[4]Discriminative training and model adaptation of RNN-LM is easier than n-gram model adaptation [222, 223, 224] and discriminative training of LM [225]

correction because of data sparsity. Our proposed method is based on a RNN-LM framework, and can consider a long context with the consideration of ASR hypotheses. Moreover, combining DLM and our discriminative RNN-LM can improve the performance from the DLM itself, which realizes short and long context discriminative language modeling.

The remainder of this section is organized as follows. Section 4.7.2 describes the conventional RNN-LM [209]. Our proposed discriminative approach is described in Section 4.7.3. Section 4.7.4 describes our experiments involving a LVCSR task and reveals that the proposed method improves speech recognition performance.

### 4.7.1 Discriminative language modeling

DLM [204, 228, 226, 227] learns patterns of errors in the $N$-best hypotheses output by a speech recognizer, and adjusts the hypotheses' scores so that the one with the least errors is selected. The score can be modified simply using the inner product of a feature vector $\boldsymbol{\phi}(s)$ extracted from a hypothesis $s$ and a weight vector $\boldsymbol{w}$. The re-scored best hypothesis $\hat{s}^{(r)}$ is then obtained as:

$$\hat{s}^{(r)} = \underset{s \in \mathcal{S}^{(r)}}{\arg\max} \left[ w^{(0)} \cdot \{p_\lambda(\boldsymbol{y}^{(r)}|\mathcal{H}_s)\}^\kappa p_L(s) + \boldsymbol{w}^\top \phi(s) \right], \tag{4.65}$$

where $w^{(0)}$ is the weight for the original acoustic and language model score, and $\mathcal{S}^{(r)}$ is the set of $N$-best hypotheses for utterance $r$. Features are usually N-gram counts. During training, separate weight vectors $\boldsymbol{w}^{(r)}$ for each speech utterance $r$ are estimated by using an on-line training algorithm, which employs the following rule:

$$\boldsymbol{w}^{(r)} \leftarrow \boldsymbol{w}^{(r-1)} + (\boldsymbol{\phi}(s^{(r)}) - \boldsymbol{\phi}(\hat{s}^{(r)})). \tag{4.66}$$

To increase the generalization ability, the weight vector used at test time is obtained by averaging the weight vectors for all training utterances [229][5]. In this section, instead of using the reference as $s^{(r)}$ in Eq. (4.66), we select $s^{(r)}$ within the $N$-best list as the hypothesis with lowest WER with respect to the reference.

### 4.7.2 RNN-LM and cross-entropy (CE) training

Fig. 4.6 shows the topology of RNN-LM having one hidden layer, which we used for the following experiments. Hidden-layer units in the previous frame are recursively connected to the input vector. Weight matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ ($\triangleq \Theta$) are model parameters to be estimated in a training phase.

[5]This is the average perceptron algorithm.

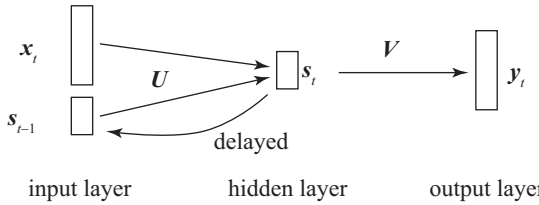



Fig. 4.6 Recurrent neural network language model (RNN-LM) topology. $|\mathcal{V}|$-dimensional input vector $\boldsymbol{x}_t$ is a 1-of-$|\mathcal{V}|$ representation of the $t$-th word of the utterance. Output vector $\boldsymbol{y}_t$ is an $|\mathcal{V}|$-dimensional posterior probability vector corresponding to input words conditioned on the previous context. The hidden layer has a low-dimensional vector $\boldsymbol{s}_t$. Hidden-layer units in the previous frame $\boldsymbol{s}_{t-1}$ are recursively concatenated to the input vector $\boldsymbol{x}_t$.

#### 4.7.2.1 CE training

We train the RNN-LM according to the CE criterion that minimizes the objective function $\mathcal{F}^{\mathrm{CE}}$. CE is calculated from a posterior of the predicted word

$$\boldsymbol{y}_t = [y_t(1), \cdots, y_t(n), \cdots, y_t(|\mathcal{V}|)]^\top \tag{4.67}$$

with vocabulary $\mathcal{V}$, and a reference label sequence $C = \{c_t | t = 1, \cdots, T\}$, as follows:

$$\mathcal{F}^{\mathrm{CE}}(C) = -\sum_{n=1}^{|\mathcal{V}|} \sum_{t=1}^{T} \delta(n, c_t) \ln y_t(n), \tag{4.68}$$

where $c_t$ is an index of the reference label at the $t$-th word. $\delta(\cdot, \cdot)$ is a Kronecker delta function. The output layer has a softmax function $y_t$:

$$y_t(n) = \frac{\exp(a_t(n))}{\sum_{n'} \exp(a_t(n'))}, \tag{4.69}$$

where $n$ is an index of elements in the output (softmax) layer and $a_t$ is an activation of the $n$-th word.

#### 4.7.2.2 Update rule

We discuss gradient-descent-based update rules for training parameter $\Theta$. Based on the chain rule property of neural network (i.e., $\partial/\partial\Theta = \partial/\partial a_t(n) \cdot \partial a_t(n)/\partial\Theta$), we focus on the differentiation of the objective function $\mathcal{F}^{\mathrm{CE}}$ w.r.t of the activation $a_t(n)$ as

$$\frac{\partial \mathcal{F}^{\mathrm{CE}}}{\partial a_t(n)} = -[\delta(n, c_t) - y_t(n)] \triangleq \varepsilon_t(n), \tag{4.70}$$

because $\partial/\partial a_t(n) \ln y_t(n') = \delta(n, n') - y_t(n)$. This equation means that the difference of the reference word and posterior $\varepsilon_t(n)$, which is an error of word $n$ at position $t$, is propagated to the estimation of the model parameters $\Theta$. Since there is a recurrent connection, it will be solved by the back propagation through time [209].

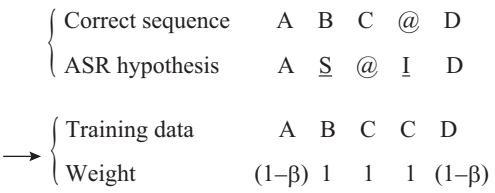



Fig. 4.7 Weight discount procedure of the proposed method. The weight of training data is discounted (i.e., $1-\beta$) for the correct data. A, B, C, and D are words, @ is a NULL token that follows the alignments of a correct word sequence and ASR hypothesis are fixed. S denotes a substitution and I denotes an insertion error. For insertion, repeated entry of the previous frame is used.

### 4.7.3 Discriminative training of RNN-LM

#### 4.7.3.1 Discriminative criterion

To introduce the discriminative training into RNN-LM, we start from the word-level likelihood ratio objective function $\mathcal{F}^{\mathrm{LR}}$[6]:

$$\mathcal{F}^{\mathrm{LR}}(C,H) = -\sum_t \ln \frac{y_t(c_t)}{y_t(h_t)^\beta}, \tag{4.71}$$

where $h_t$ is an index of the $t$-th word of the 1-best ASR hypothesis aligned with the reference sequence $C$, and $H = \{h_t | t = 1, \cdots, T\}$ denotes the 1-best ASR sequence. $\beta$ is a scaling factor, and the meaning of this factor will be discussed later. Note that this log likelihood ratio has a property of a discriminative criterion (used in Minimum classification error (MCE) training [230, 147, 231] [7] and DLM [204]) so that minimizing $\mathcal{F}^{\mathrm{LR}}(C,H)$ corresponds to correct misrecognized $h_t$ approaches to reference $c_t$.

Equation (4.71) can also be rewritten as

$$\mathcal{F}^{\mathrm{LR}}(C,H) = -\sum_n \sum_t \delta(n,c_t)\ln y_t(n) - \beta\delta(n,h_t)\ln y_t(n) = \mathcal{F}^{\mathrm{CE}}(C) - \beta\mathcal{F}^{\mathrm{CE}}(H). \tag{4.72}$$

Therefore, Equation (4.71) can be interpreted as a weighted difference of CE for the correct label and ASR hypothesis.

#### 4.7.3.2 Update rule

For our proposed model, the update rule corresponds to (4.70) is also derived from the differentiation of (4.72) such that

$$\frac{\partial \mathcal{F}^{\mathrm{LR}}(C,H)}{\partial a_t(n)} = -[\delta(n,c_t) - \beta\delta(n,h_t) - (1-\beta)y_t(n)]. \tag{4.73}$$

[6] This is not a sequence discriminative training but a word-level discriminative training based on an alignment between reference and 1-best ASR hypothesis.

[7] We can also consider an MMI-type discriminative criterion by summing up all possible hypotheses in the denominator.

In our implementation, we assume $(1-\beta)y_t(n)$ as $y_t(n)$ for simplicity, thus we obtain

$$\frac{\partial \mathcal{F}^{\mathrm{LR}}(C,H)}{\partial a_t(n)} \approx -[\delta(n,c_t) - \beta\delta(n,h_t) - y_t(c_t)]. \tag{4.74}$$

Fig. 4.7 shows a weight discount of the proposed method. First, alignments of correct word sequences and ASR hypotheses are fixed using dynamic programming. Second, the weight for the correct label is discounted (i.e., $1-\beta$) and the model is re-trained with these discounted weights. Note that we assume that $\delta(n,c_t) - \beta\delta(n,h_t) = 0$ when $\delta(n,c_t) - \beta\delta(n,h_t) < 0$ to avoid that the value of target reference word becomes negative.

#### 4.7.3.3 Word-level confidence measure

Word-level confidence measure $\nu_t$ $(0 \leq \nu_t \leq 1)$, which is calculated from a confusion network, can be used to adjust the discount factor $\beta$. Errors with high confidence are more problematic and should be weighted more than errors with low confidence. Equation (4.74) is modified as follow.

$$\frac{\partial \mathcal{F}^{\mathrm{LR}}(C,H)}{\partial a_t(n)} = -[\delta(n,c_t) - \beta(1-\nu_t(h_t))\delta(n,h_t) - y_t(c_t)]. \tag{4.75}$$

Thus, we can control the discount value according to the confidence in the update rule.

#### 4.7.3.4 Smoothing with original CE model

Finally, RNN-LM models are obtained by smoothing parameters obtained by the proposed discriminative method $\boldsymbol{U}^{\mathrm{LR}}, \boldsymbol{V}^{\mathrm{LR}}$ with the original CE model $\boldsymbol{U}^{\mathrm{CE}}, \boldsymbol{V}^{\mathrm{CE}}$ such that

$$\{\boldsymbol{U},\boldsymbol{V}\} \leftarrow \tau\{\boldsymbol{U}^{\mathrm{CE}},\boldsymbol{V}^{\mathrm{CE}}\} + (1-\tau)\{\boldsymbol{U}^{\mathrm{LR}},\boldsymbol{V}^{\mathrm{LR}}\}, \tag{4.76}$$

where $\tau$ is a smoothing factor. This avoids over-training.

### 4.7.4 Experimental setups

We evaluated the observed performance improvement on the CSJ (Section 4.3.3). Vocabulary size is about 70k. We used three types of test sets wherein each set consists of lecture-style examples from 10 speakers. Test sets E1, E2, and E3 contain 22,682, 23,226, and 14,896 words, respectively.

We trained the DNN-HMM with CE training using 23 dimensional mel-filter bank coefficients + $\Delta$+$\Delta\Delta$. The number of context-dependent HMM states was 3,500 and the DNN contained seven hidden layers and 2,048 nodes per layer in accordance with settings used in a previous study [153]. The initial learning rate was 0.01 and decreased to 0.001 at the end of training. After a CE DNN acoustic model was obtained, boosted MMI discriminative training for DNN [185] was conducted. We used nnet2 implementation of DNN training tools in a Kaldi toolkit [124].

Table 4.15 WER [%] on CSJ using a DNN acoustic model with a conventional $n$-gram and discriminative language model (DLM).

| | E1 | E2 | E3 | Avg. |
|---|---|---|---|---|
| baseline | 12.81 | 10.64 | 11.13 | 11.53 |
| +DLM | 12.60 | 10.52 | 10.82 | 11.31 |

Table 4.16 WER [%] on CSJ using a DNN acoustic model with RNN-LM-based and DLM-based rescoring.

| | E1 | E2 | E3 | Avg. |
|---|---|---|---|---|
| +RNN-LM | 11.97 | 10.18 | 10.51 | 10.89 |
| +RNN-LM+DLM | 11.74 | 9.98 | 10.03 | 10.58 |

Although the size of the original language model was 70k, the vocabulary size of RNN-LM was limited to 10k, which corresponds to the number of input layer dimensions (i.e., $|\mathcal{V}|$). The number of hidden-layer units was 30. The learning rate for RNN-LM, $\eta$, was 0.1 or 0.05. RNN-LM was constructed using the RNN-LM toolkit [211]. The language model score was obtained by linear interpolation of the RNN-LM score and the original n-gram model score. The weight of interpolation was 0.5 and 100-best hypotheses for each utterance were used for rescoring. We combined the RNN-LM and the proposed discriminative RNN-LM with DLM.

### 4.7.5 Results and discussion

#### 4.7.5.1 Baseline results

Table 4.15 shows the baseline results when using the discriminatively trained DNN acoustic model, which was state-of-the-art performance for this CSJ corpus [226, 153]. Using DLM rescoring, the word error rate (WER) was improved by 0.22% on average.

For this high baseline, RNN-LM rescoring significantly improved the WER, as shown in Table 4.16 by 0.64% on average. In addition to RNN-LM, the DLM was also effective for this result, which shows the effectiveness of the discriminative model.

#### 4.7.5.2 Proposed method

Table 4.17 shows the proposed discriminative RNN-LM (d-RNN-LM). Three parameters exist in the prosed method and parametric studies were conducted. In nearly all cases, average WER was better than that of the RNN-LM result in Table 4.16. This result suggest that the parameter tuning was not so difficult. Table 4.18 shows that DLM was effective when used with the proposed method because the explicit use of short context by the $n$-gram model was powerful whereas the proposed method implicitly used short context.

Table 4.17 WER [%] on CSJ with the proposed discriminative RNN-LM (d-RNN-LM).

| $\beta$ | $\tau$ | $\eta$ | E1 | E2 | E3 | Avg. |
|---|---|---|---|---|---|---|
| 0.05 | 0.85 | 0.1 | 11.99 | 10.19 | 10.50 | 10.89 |
| | | 0.05 | **11.84** | 10.07 | 10.61 | 10.84 |
| | 0.9 | 0.1 | 11.91 | **10.02** | 10.51 | 10.81 |
| | | 0.05 | **11.84** | 10.03 | 10.49 | 10.79 |
| 0.10 | 0.85 | 0.1 | 12.20 | 10.45 | 10.69 | 11.11 |
| | | 0.05 | 11.86 | 10.09 | 10.47 | 10.81 |
| | 0.9 | 0.1 | 11.93 | 10.19 | 10.41 | 10.84 |
| | | 0.05 | 11.90 | 10.04 | **10.39** | **10.78** |
| 0.15 | 0.85 | 0.1 | 12.06 | 10.38 | 10.49 | 10.98 |
| | | 0.05 | 11.93 | 10.09 | 10.40 | 10.81 |
| | 0.9 | 0.1 | 11.98 | 10.17 | **10.39** | 10.85 |
| | | 0.05 | 11.98 | 10.03 | **10.39** | 10.80 |

Table 4.18 WER [%] on CSJ with the proposed discriminative RNN-LM (d-RNN-LM) and DLM rescoring.

| $\beta$ | $\tau$ | $\eta$ | E1 | E2 | E3 | Avg. |
|---|---|---|---|---|---|---|
| 0.05 | 0.85 | 0.1 | 12.00 | 10.20 | 10.51 | 10.90 |
| | | 0.05 | 11.68 | 9.98 | 10.04 | 10.57 |
| | 0.9 | 0.1 | 11.72 | 10.01 | 10.04 | 10.59 |
| | | 0.05 | 11.63 | **9.90** | 10.05 | 10.53 |
| 0.10 | 0.85 | 0.1 | 12.07 | 10.19 | 10.70 | 10.99 |
| | | 0.05 | 11.75 | 10.03 | 10.28 | 10.69 |
| | 0.9 | 0.1 | 11.77 | 10.03 | 10.12 | 10.64 |
| | | 0.05 | 11.64 | 9.94 | 10.08 | 10.55 |
| 0.15 | 0.85 | 0.1 | 11.81 | 10.07 | 10.26 | 10.71 |
| | | 0.05 | 11.63 | 10.00 | 10.14 | 10.59 |
| | 0.9 | 0.1 | 11.61 | 9.95 | 10.01 | 10.52 |
| | | 0.05 | **11.60** | 9.95 | **9.99** | **10.51** |

Table 4.19 shows the proposed method using word-level confidence measures. Unfortunately, little performance gain was observed, but similar tendencies were noticeable. DLM was also effective as shown in Table 4.20.

Although the performance gain of the proposed method was small in our experiments overall, this is simply due to very high baseline of this setting. We believe that this modeling increases model estimation robustness for a task that contains many errors.

### 4.7.6 Conclusion

We proposed a discriminative training method for RNN-LM. In addition to the CE training of correct examples, discriminative training against ASR hypotheses was proposed. The proposed discriminative training yielded a difference of CE that was similar to the difference statistics revealed in the discriminative training of acoustic modeling. Experimental results showed that

Table 4.19 WER [%] on CSJ with the proposed discriminative RNN-LM (d-RNN-LM) using word-level confidence measures.

| $\beta$ | $\tau$ | $\eta$ | E1 | E2 | E3 | Avg. |
|---|---|---|---|---|---|---|
| 0.05 | 0.85 | 0.1 | 12.15 | 10.34 | 10.49 | 10.99 |
| | | 0.05 | 11.88 | 10.08 | 10.54 | 10.83 |
| | 0.9 | 0.1 | 11.93 | 10.17 | 10.44 | 10.85 |
| | | 0.05 | **11.84** | **10.03** | 10.46 | **10.78** |
| 0.10 | 0.85 | 0.1 | 12.39 | 10.43 | 10.92 | 11.25 |
| | | 0.05 | 11.89 | 10.13 | 10.52 | 10.85 |
| | 0.9 | 0.1 | 12.02 | 10.17 | 10.51 | 10.90 |
| | | 0.05 | 11.93 | 10.09 | 10.35 | 10.79 |
| 0.15 | 0.85 | 0.1 | 12.18 | 10.41 | 10.60 | 11.06 |
| | | 0.05 | 11.95 | 10.11 | **10.31** | 10.79 |
| | 0.9 | 0.1 | 12.01 | 10.21 | 10.45 | 10.89 |
| | | 0.05 | 11.95 | 10.04 | 10.35 | **10.78** |

Table 4.20 WER [%] on CSJ with the proposed discriminative RNN-LM (d-RNN-LM) using word-level confidence measures and DLM rescoring.

| $\beta$ | $\tau$ | $\eta$ | E1 | E2 | E3 | Avg. |
|---|---|---|---|---|---|---|
| 0.05 | 0.85 | 0.1 | 11.84 | 10.28 | 10.41 | 10.84 |
| | | 0.05 | **11.56** | 10.03 | 10.12 | 10.57 |
| | 0.9 | 0.1 | 11.66 | 9.99 | 10.11 | 10.59 |
| | | 0.05 | 11.63 | 9.94 | 10.06 | 10.54 |
| 0.10 | 0.85 | 0.1 | 11.71 | 10.00 | 10.22 | 10.64 |
| | | 0.05 | 11.65 | 10.02 | 10.27 | 10.65 |
| | 0.9 | 0.1 | 11.65 | 9.95 | 10.19 | 10.60 |
| | | 0.05 | 11.66 | 9.94 | 10.14 | 10.58 |
| 0.15 | 0.85 | 0.1 | 11.76 | 10.04 | 10.18 | 10.66 |
| | | 0.05 | 12.01 | 10.20 | 10.57 | 10.93 |
| | 0.9 | 0.1 | 11.63 | **9.84** | **10.01** | **10.49** |
| | | 0.05 | 11.69 | 9.99 | 10.14 | 10.61 |

our proposed method improved the performance of an LVCSR task. Combining the proposed discriminative RNN-LM, which uses short and long context implicitly, and the DLM, which uses short context explicitly, was also effective because the two complement one another. Future research will examine sequential discriminative training and the use of N-best hypotheses in the training.

## 4.8 Uncertainty training and decoding methods of DNN

Recently, several methods that were developed for the GMM have been applied to DNNs. For example, fMLLR, an effective speaker-adaptation technique, is widely used for as the DNN front-end [232]. This section applies uncertainty techniques to DNNs because uncertainty techniques are successful examples in noisy ASR for GMM-based systems.

In noisy condition, speech enhancement improves the ASR performance, even for a DNN-based systems [233, 234, 235]. However, distortions are consequently introduced to the speech, and this can degrade the ASR performance. This is problematic especially when noise conditions are mismatched between training and decoding time, or when speech enhancement is only applied during decoding, because mismatches of the acoustic model or speech distortion significantly degrades ASR performance.

To address this problem, several methods have been proposed to adjust features according to their reliabilities representing the distortion by speech enhancement. For the GMM, starting from the missing data theory [236, 237, 238], when feature uncertainty can be represented as a Gaussian distribution, the GMM likelihoods are computed based on the expectations with respect to these feature-uncertainty distributions. The expectation is calculated analytically by integrating out marginal parameters, and this marginalization renders models more robust to speech distortions caused by speech enhancement, and it is referred to as the uncertainty-decoding technique. As a result, covariance matrices for the Gaussian distributions of the acoustic models for input features are adjusted corresponding to the extent of uncertainties (i.e., reliability). Many uncertainty methods have been proposed, and their effectiveness for the GMM has been demonstrated experimentally [239, 240, 241, 242, 243, 244, 245]. For example, [241, 242] used a difference vector between noisy and enhanced feature vectors, [243] used a posterior variance of Wiener filters, and [246] used an estimate based on a binary speech/noise predominance model. However, because of an inclusion of non-linear activations in DNNs, it is difficult to handle uncertainty propagations analytically.

This section proposes uncertainty training and decoding methods for DNNs. Unlike [247], which calculates the expectation operation approximately for the DNN score calculation and for training of DNNs, our method samples some input features based on uncertainties by using the Monte-Carlo method. However, because DNN model training requires considerable computation, efficient sampling is essential. The proposed method focuses on interpolation vectors before and after speech enhancement, and it efficiently represents the feature distributions of enhanced speech vectors by sampling interpolation coefficients probabilistically. In addition, sampling is also performed for decoding, and multiple recognition hypotheses for each sample are combined to further improve the performance.

The theory behind the uncertainty technique is based on the following conditional expectation operation:

$$\mathbb{E}[f(\boldsymbol{y}_{1:T})|\boldsymbol{x}_{1:T}] \triangleq \int f(\boldsymbol{y}_{1:T})p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})d\boldsymbol{y}_{1:T}, \tag{4.77}$$

where $\boldsymbol{x}_{1:T} = \{\boldsymbol{x}_t|t = 1, \ldots, T\}$ is a sequence of $T$ noisy feature vector and $\boldsymbol{y}_{1:T}$ is a sequence

of $T$ enhanced features. $f()$ denotes decoding (see Section 4.8.1) or training (see Section 4.8.2) depending on the application of our target[8]. $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})$ is a stochastic representation of an enhanced feature sequence with its uncertainty (see Section 4.8.3).

### 4.8.1 DNN uncertainty decoding

We first focus on uncertainty decoding for DNNs with a hybrid architecture that combines the hidden Markov model (HMM) with the DNN. In this framework, $f()$ in Eq. (4.77) is represented by the following actual decoding process:

$$\hat{W} = \mathbb{E}\left[\arg\max_{W} p(\boldsymbol{y}_{1:T}|\mathcal{H}_W)p(W)\middle|\boldsymbol{x}_{1:T}\right] = \mathbb{E}\left[W_{\boldsymbol{y}_{1:T}}|\boldsymbol{x}_{1:T}\right], \tag{4.78}$$

where $W$ is a word sequence and $\mathcal{H}_W$ is a possible HMM state-sequence given $W$. $W_{\boldsymbol{y}_{1:T}}$ is a decoded word sequence given input feature sequence $\boldsymbol{y}_{1:T}$. Note that some conventional uncertainty techniques based on the GMM provide an analytical solution to Eq. (4.78) by integrating out the expectation operations for $\mathbb{E}[p(\boldsymbol{y}_{1:T}|\mathcal{H}_W)|\boldsymbol{x}_{1:T}]$ with a Gaussian-based uncertainty for $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})$ (see [248] for more details). However, DNN-based acoustic models cannot obtain such analytical solutions, owing to the presence of nonlinear activation functions; these models require approximations [247, 249].

Rather than using approximations, we adopt a straightforward expectation from Eq. (4.78), based on a Monte-Carlo sampling, and averaging out multiple outputs at the hypothesis level rather than integrals. These outputs are obtained from decoding processes with different feature samples. The disadvantage to this approach is that it requires the ASR decoding computations for all samples, even though lattice re-scoring can decrease these computations. In addition, it is very difficult to sample $\boldsymbol{y}_{1:T}$ to fully cover a possible input feature space. Instead of directly considering the distribution of sequential input feature $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T})$, we assume a deterministic relationship for the sampled input feature $\boldsymbol{y}_t$ at the frame $t$ based on a linear interpolation between $\boldsymbol{x}_t$ and $\hat{\boldsymbol{y}}_t$ as:

$$\boldsymbol{y}_t = \hat{\boldsymbol{y}}_t + \alpha(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t) \text{ for } t = 1, \ldots, T, \tag{4.79}$$

where $\alpha$ is a linear interpolation coefficient. The geometric meaning of this linear interpolation is shown in Fig. 4.8. This approach is inspired by uncertainty decoding based on an approximated observation distribution with the covariance matrix obtained by the difference between noisy and enhanced features: $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T}) \approx \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{y}_t|\hat{\boldsymbol{y}}_t, \left[\alpha(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t)(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t)^\top\right])$ in [241, 242]. In fact, Eq. (4.79) can be regarded as a sigma point for this distribution [250]. Then, we regard the linear interpolation coefficient $\alpha$ as a random variable, and efficiently sample one-dimensional $\alpha$ with a relatively small number of samples.

Thus, our proposed uncertainty decoding with $N$ Monte Carlo samples is represented from Eq. (4.78) as follows:

$$\hat{W} = R\left[\left\{W_{\boldsymbol{y}^n_{1:T}}\right\}_{n=1}^{N}\right], \quad \boldsymbol{y}^n_t = \hat{\boldsymbol{y}}_t + \alpha^n(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t) \text{ for } t = 1, \ldots, T, \ \alpha^n \sim p(\alpha), \tag{4.80}$$

[8] Although $f()$ has several options including an acoustic score function [248], this section regards $f()$ as an entire decoding process, which returns output sequences.

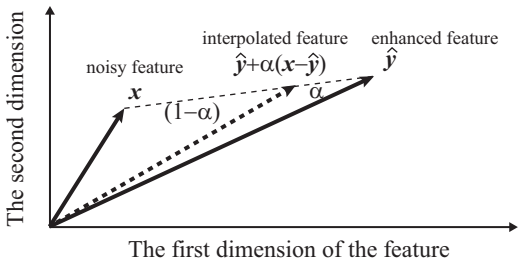



Fig. 4.8 Noisy feature $\boldsymbol{x}$ and enhanced feature $\hat{\boldsymbol{y}}$, and the sampling of feature $\boldsymbol{y}$ based on an interpolation between them.

where $R[\cdot]$ is performed by using a hypothesis-level integration, e.g., with Recognizer Output Voting Error Reduction (ROVER) [195]. $\alpha^n \sim p(\alpha)$ means that the $n$-th $\alpha$ is sampled from the distribution $p(\alpha)$. Section 4.8.3 discusses $p(\alpha)$ in more detail.

### 4.8.2 DNN uncertainty training

In a manner similar to the description in Section 4.8.1, uncertainty training, given a reference word sequence $W$, can be represented by replacing $f()$ in Eq. (4.77) with a training procedure:

$$\hat{\Theta} = \mathbb{E}\left[\arg\min_{\Theta} \mathcal{F}_{\Theta}(\boldsymbol{y}_{1:T}, W) \middle| \boldsymbol{x}_{1:T}\right], \tag{4.81}$$

where $\mathcal{F}_{\Theta}$ is an objective function of the DNN, e.g., cross entropy (CE) or sequence-discriminative criteria, with the model parameter $\Theta$.

The input features are sampled based on the distribution of a linear interpolation coefficient $p(\alpha)$ similarly to the proposed uncertainty decoding in Section 4.8.1. Instead of the expectation operation with respect to parameters in Eq. (4.81), we propose to use a Monte Carlo sampling for an objective function

$$\hat{\Theta} = \arg\min_{\Theta} \mathbb{E}\left[\mathcal{F}_{\Theta}(\boldsymbol{y}_{1:T}, W) | \boldsymbol{x}_{1:T}\right] \approx \arg\min_{\Theta} \sum_{n=1}^{N} \mathcal{F}_{\Theta}(\boldsymbol{y}_{1:T}^n, W), \tag{4.82}$$

where $\boldsymbol{y}_t^n = \hat{\boldsymbol{y}}_t + \alpha^n(\boldsymbol{x}_t - \hat{\boldsymbol{y}}_t) \forall t, \quad \alpha^n \sim p(\alpha)$. For CE training, the objective function with the Monte Carlo sampling is represented as follows:

$$\sum_{n=1}^{N} \mathcal{F}_{\Theta}^{\mathrm{CE}}(\boldsymbol{y}_{1:T}^n, W) = -\sum_{t=1}^{T}\sum_{n=1}^{N} \ln p_{\Theta}(s_t | \boldsymbol{y}_t^n), \tag{4.83}$$

where $s_t$ is an HMM state at the frame $t$, obtained by the Viterbi alignment given $W$. Thus, the additivity to the objective function enables the expectation operation, simply by using the sampled training data as input features. This approach can also be applied to sequence-discriminative DNN training, e.g., [185]. The proposed approach is motivated by a deep learning method, which has recently been used in the area of image processing [251, 252] to train DNN models by sampling input features based on possible feature changes. Such an approach renders models robust and invariant to these changes.

### 4.8.3 Stochastic process for the linear-interpolation coefficient

We sample multiple $\alpha$'s for each utterance by using the following one-dimensional Gaussian mixture with $K$ mixture components to sample $\alpha$:

$$p(\alpha) = \sum_{k=1}^{K} w_k \mathcal{N}(\alpha|\mu_k, \sigma), \tag{4.84}$$

where the mean $\mu_k$ is empirically determined from some values in $[0, 1]$, so that the input feature $\boldsymbol{y}_t$ is sampled between the noisy feature $\boldsymbol{x}_t$ and the enhanced feature $\hat{\boldsymbol{y}}_t$. The variance $\sigma$ and the mixture weight $w_k(= 1/K)$ are fixed, and in some experiments $\alpha \in \{\mu_k\}_{k=1}^{K}$ are fixed, i.e., $\sigma \to 0$.

### 4.8.4 Experimental setups

We validated the effectiveness of our proposed approaches with two noisy and reverberated ASR tasks. The first corpus was the second CHiME challenge Track 2 (details are in Section 5.2). The MNMF algorithm [253, 86] was used for speech enhancement. The second corpus was the REVERB challenge simulation data (details are in Section 5.5). Multi-channel BF with DOA estimation and a single-channel dereveberation were applied (Sections 5.5.3 and 3.2).

The ASR settings were the same for both tasks. Some tuning parameters, e.g., language model weights, were optimized based on the word error rate (WER) of the development set. The vocabulary size was 5k and a trigram language model was used. These systems were constructed using the Kaldi toolkit [124]. The learning rates were reduced for the proposed uncertainty-training method, because the interpolated training data were similar to the original data and acoustic models tend to be overly tuned. We used 40-dimensional filter bank features with $\Delta$ and $\Delta\Delta$. The DNN acoustic models were constructed according to the CE criterion before performing sequential minimum Bayes risk (SMBR) discriminative training [185].

The following six system types were prepared.

(1) noisy: decoding $\boldsymbol{x}$ (trained on $\boldsymbol{x}$)

(2) enhan (enhanced): decoding $\hat{\boldsymbol{y}}$ (trained on $\boldsymbol{y}$)

(3) diff (difference): decoding $\left[\hat{\boldsymbol{y}}^\top, [\boldsymbol{x} - \hat{\boldsymbol{y}}]^\top\right]^\top$

(4) uncert(t) (uncertainty training): decoding $\hat{\boldsymbol{y}}$, whereas models were trained on $\hat{\boldsymbol{y}} + \alpha[\boldsymbol{x} - \hat{\boldsymbol{y}}]$ with $\mu_k \in \{0, 0.1, 0.2\}$.

(5) uncert(d) (uncertainty decoding): decoding $\hat{\boldsymbol{y}} + \alpha[\boldsymbol{x} - \hat{\boldsymbol{y}}]$ with $\mu_k \in \{0, 0.1, 0.2\}$, whereas models were trained on $\hat{\boldsymbol{y}}$. Their hypotheses were combined using ROVER.

(6) uncert(t,d) (combination of uncertainty training and decoding): decoding $\hat{\boldsymbol{y}} + \alpha[\boldsymbol{x} - \hat{\boldsymbol{y}}]$ with $\mu_k \in \{0, 0.1, 0.2\}$, and models were trained with the same features. Their hypotheses were also combined using ROVER.

Table 4.21 WER [%] on the development set of the second CHiME challenge (Track 2).

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| *CE | | | | | | | |
| noisy | 51.03 | 39.59 | 32.17 | 26.11 | 21.71 | 18.88 | 31.58 |
| enhan | 42.79 | 33.91 | 28.71 | 23.32 | 20.83 | 17.76 | 27.89 |
| diff | 43.19 | 34.21 | 27.75 | 23.12 | 20.30 | 17.39 | 27.66 |
| uncert(t) | 42.29 | 32.87 | 27.63 | 22.27 | 20.68 | 17.10 | 27.14 |
| uncert(d) | 42.19 | 33.22 | 28.37 | 23.38 | 20.43 | 17.55 | 27.52 |
| uncert(t,d) | 41.92 | 32.60 | 27.48 | 22.13 | 20.64 | 17.02 | **26.97** |
| *SMBR | | | | | | | |
| noisy | 48.05 | 36.64 | 29.18 | 23.60 | 18.90 | 17.01 | 28.90 |
| enhan | 39.15 | 30.95 | 24.99 | 20.36 | 18.54 | 15.50 | 24.92 |
| diff | 39.42 | 30.46 | 24.35 | 20.56 | 17.47 | 15.39 | 24.61 |
| uncert(t) | 37.90 | 30.64 | 24.55 | 20.40 | 17.57 | 15.19 | 24.37 |
| uncert(d) | 38.50 | 30.05 | 24.58 | 20.30 | 18.31 | 15.49 | 24.54 |
| uncert(t,d) | 37.04 | 29.72 | 24.19 | 19.78 | 16.98 | 15.08 | **23.80** |

## 4.8.5 Results and discussion

### 4.8.5.1 The second CHiME challenge: Track 2

Table 4.21 shows the WER from the second CHiME challenge development set. Speech enhancement by MNMF significantly improved the ASR performance of the DNN system. Concatenating difference features ("diff" in table, this is motivated by [241, 242] but it simply stacks uncertainty observations) to input features reduced the WER for the CE model by 0.23%, and by 0.31% for the SMBR (discriminatively trained) model. This experiment used fixed $\alpha$'s, i.e., $\alpha \in \{0, 0.1, 0.2\}$ ($\sigma \to 0$ in Eq. (4.84)). The proposed uncertainty decoding ("uncert(d)" in the table) reduced the WER by 0.37% and 0.38% for the CE and SMBR models, respectively. In this case, model re-training was unnecessary but the computational time increased for decoding. The proposed uncertainty training ("uncert(t)") reduced the WER by 0.75% and 0.55% for the CE and SMBR models, respectively. In this case, training time increased, whereas the decoding time was almost the same as it was for "enhan" and "diff". For the DNN acoustic models, it is more effective to consider uncertainties for training than for decoding. When uncertainties are introduced to both training and decoding ("uncert(t,d)"), the WERs were significantly improved, by 0.92% and 1.12%, for the CE and SMBR models, respectively.

Table 4.22 shows the effectiveness of random perturbation ($\sigma > 0$ in Eq.(4.84)) to the interpolated points (see Section 4.8.3). Although, for all $\sigma$'s, this method did not improve the ASR performance for uncertainty decoding ("uncert(d)"), it improved the performance for both uncertainty training ("uncert(t)") and the combination of training with decoding ("uncert(t,d)"). In the case of $\sigma = 0.015$, for the CE acoustic model, the WER improved by 0.31% for training, and by 0.71% for the combination of training with decoding. However, this method did not improve the ASR performance for the SMBR model, which is robust to frequent error patterns.

Table 4.23 shows the WER on the evaluation set, where '+p' denotes the case of $\sigma = 0.015$.

Table 4.22 WER [%] on development set of the second CHiME challenge with the addition of random perturbation to the interpolated points.

| $\sigma$ | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| *CE | | | | | | | |
| uncert(t) | | | | | | | |
| 0 | 42.29 | 32.87 | 27.63 | 22.27 | 20.68 | 17.10 | 27.14 |
| 0.005 | 41.45 | 32.13 | 27.39 | 22.92 | 20.15 | 17.10 | 26.86 |
| 0.010 | 41.70 | 32.44 | 27.51 | 22.76 | 20.19 | 17.30 | 26.99 |
| 0.015 | 41.08 | 32.76 | 27.63 | 23.01 | 19.81 | 16.65 | **26.83** |
| uncert(d) | | | | | | | |
| 0 | 42.19 | 33.22 | 28.37 | 23.38 | 20.43 | 17.55 | **27.52** |
| 0.005 | 42.23 | 33.19 | 28.46 | 23.37 | 20.42 | 17.54 | 27.53 |
| 0.010 | 42.26 | 33.22 | 28.53 | 23.37 | 20.42 | 17.58 | 27.56 |
| 0.015 | 42.26 | 33.22 | 28.54 | 23.34 | 20.40 | 17.60 | 27.56 |
| uncert(t,d) | | | | | | | |
| 0 | 41.92 | 32.60 | 27.48 | 22.13 | 20.64 | 17.02 | 26.97 |
| 0.005 | 40.85 | 31.73 | 27.13 | 22.82 | 19.80 | 16.79 | 26.52 |
| 0.010 | 40.60 | 32.04 | 26.94 | 22.17 | 19.59 | 17.20 | 26.42 |
| 0.015 | 40.54 | 31.82 | 27.36 | 22.33 | 19.13 | 16.36 | **26.26** |
| *SMBR | | | | | | | |
| uncert(t) | | | | | | | |
| 0 | 37.90 | 30.64 | 24.55 | 20.40 | 17.57 | 15.19 | **24.37** |
| 0.005 | 38.40 | 30.40 | 24.86 | 20.21 | 18.03 | 15.34 | 24.54 |
| 0.010 | 38.72 | 30.45 | 25.53 | 20.73 | 17.41 | 15.36 | 24.70 |
| 0.015 | 38.03 | 31.02 | 25.74 | 21.48 | 17.85 | 15.64 | 24.95 |
| uncert(d) | | | | | | | |
| 0 | 38.50 | 30.05 | 24.58 | 20.30 | 18.31 | 15.49 | 24.54 |
| 0.005 | 38.44 | 30.08 | 24.58 | 20.31 | 18.31 | 15.49 | **24.53** |
| 0.010 | 38.49 | 29.89 | 24.71 | 20.39 | 18.01 | 15.70 | **24.53** |
| 0.015 | 38.49 | 30.20 | 24.55 | 20.19 | 18.29 | 15.49 | **24.53** |
| uncert(t,d) | | | | | | | |
| 0 | 37.04 | 29.72 | 24.19 | 19.78 | 16.98 | 15.08 | **23.80** |
| 0.005 | 37.72 | 30.33 | 24.34 | 20.08 | 17.27 | 15.30 | 24.17 |
| 0.010 | 37.69 | 29.84 | 24.83 | 20.24 | 17.01 | 15.08 | 24.11 |
| 0.015 | 37.00 | 30.09 | 24.84 | 20.64 | 17.39 | 15.50 | 24.24 |

In this case, the introduction of uncertainties improved the performance of training more than decoding, and it achieved the best performance in the case of "uncert(t,d)". This trend was similar to that of the development set. In this case, random perturbation to the uncertainty training and both training and decoding improved the performance even for the SMBR model. This shows that perturbation renders the acoustic models more robust to unknown data. Finally, the proposed method reduced the WER from "enhan" for the CE model by 1.13% and for SMBR model by 0.43%, and outperformed the "diff" by 0.12% and −0.41%. These results confirmed the effectiveness of the proposed method.

Table 4.23 WER [%] on the evaluation set of the second CHiME challenge, where '+p' refers to the inclusion of random perturbation at $\sigma = 0.015$.

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| *CE | | | | | | | |
| noisy | 44.07 | 34.56 | 28.40 | 20.46 | 17.13 | 14.72 | 26.56 |
| enhan | 36.56 | 27.65 | 23.50 | 19.33 | 16.46 | 15.04 | 23.09 |
| diff | 38.05 | 28.58 | 23.13 | 18.85 | 15.62 | 13.58 | 22.97 |
| uncert(t) | 35.57 | 27.03 | 22.57 | 19.50 | 15.54 | 14.18 | 22.40 |
| +p | 35.62 | 27.29 | 22.53 | 18.27 | 15.77 | 13.77 | 22.21 |
| uncert(d) | 35.98 | 27.27 | 23.31 | 19.02 | 15.97 | 14.59 | 22.69 |
| +p | 35.94 | 27.26 | 23.28 | 18.98 | 16.01 | 14.65 | 22.69 |
| uncert(t,d) | 35.23 | 26.51 | 22.36 | 19.15 | 15.24 | 14.16 | 22.11 |
| +p | 35.16 | 26.62 | 22.42 | 18.48 | 15.62 | 13.49 | **21.96** |
| *SMBR | | | | | | | |
| noisy | 40.91 | 32.21 | 26.42 | 18.64 | 15.54 | 13.82 | 24.59 |
| enhan | 32.11 | 25.22 | 20.49 | 16.74 | 14.46 | 12.72 | 20.29 |
| diff | 33.44 | 25.95 | 20.83 | 17.04 | 14.33 | 12.65 | 20.70 |
| uncert(t) | 32.36 | 25.82 | 20.68 | 17.17 | 14.16 | 12.91 | 20.51 |
| +p | 31.40 | 25.18 | 20.85 | 17.58 | 14.50 | 12.78 | 20.38 |
| uncert(d) | 31.89 | 24.64 | 20.16 | 16.59 | 14.22 | 12.42 | 19.99 |
| +p | 31.85 | 24.79 | 20.19 | 16.50 | 14.22 | 12.44 | 20.00 |
| uncert(t,d) | 31.98 | 24.68 | 20.31 | 17.15 | 13.92 | 12.57 | 20.10 |
| +p | 30.66 | 24.73 | 20.38 | 17.07 | 13.97 | 12.35 | **19.86** |

#### 4.8.5.2 The REVERB challenge

Table 4.24 shows the WER on the development set of the REVERB challenge. The experiments in this section used fixed $\alpha$'s. Although the baseline performance was better than it was with the CHiME challenge, the proposed method was also effective and the trends were similar, i.e., the proposed method was more effective for training than decoding, and the combination further improved the performance.

Table 4.25 shows the WER on the evaluation set. The proposed method improved the WER from "enhan" for CE model by 0.52% and for SMBR model by 0.15%, and outperformed "diff" by 0.13% and −0.01%. Thus, the proposed method improved the ASR performance for two tasks.

### 4.8.6 Conclusion

This section proposed uncertainty training and decoding methods for DNN acoustic models to address observation uncertainties caused by speech enhancement. Our proposed method did not change the structure or the training and decoding strategy of the DNN. Rather, it realized uncertainty training and decoding with an efficient sampling method for enhanced features. By comparing the introduction of uncertainties to training and decoding, we discovered that the introduction of uncertainty to the training is the most effective. In addition, a random perturbation of interpolated points further improved the performance. The effectiveness of the

Table 4.24 WER [%] on the development set of the REVERB challenge simulation data.

| | Room1 | | Room2 | | Room3 | | Avg. |
|---|---|---|---|---|---|---|---|
| | far | near | far | near | far | near | |
| *CE | | | | | | | |
| noisy | 6.69 | 5.16 | 11.17 | 7.02 | 13.18 | 8.14 | 8.56 |
| enhan | 6.78 | 5.85 | 9.86 | 6.11 | 10.36 | 6.97 | 7.66 |
| diff | 6.15 | 5.01 | 9.69 | 6.21 | 9.82 | 6.28 | 7.19 |
| uncert(t) | 6.59 | 5.53 | 9.29 | 5.92 | 9.77 | 6.13 | 7.21 |
| uncert(d) | 6.74 | 5.68 | 9.93 | 6.11 | 10.44 | 6.95 | 7.64 |
| uncert(t,d) | 6.44 | 5.43 | 9.17 | 6.09 | 9.77 | 5.98 | **7.15** |
| *SMBR | | | | | | | |
| noisy | 5.36 | 4.11 | 9.54 | 5.52 | 10.29 | 6.90 | 6.95 |
| enhan | 5.51 | 4.57 | 7.79 | 5.13 | 8.21 | 5.04 | 6.04 |
| diff | 5.29 | 4.20 | 7.96 | 5.20 | 7.72 | 5.37 | 5.96 |
| uncert(t) | 5.41 | 4.30 | 7.42 | 5.15 | 8.11 | 4.77 | 5.86 |
| uncert(d) | 5.26 | 4.62 | 7.59 | 4.95 | 8.33 | 5.46 | 6.04 |
| uncert(t,d) | 5.29 | 4.18 | 7.54 | 5.15 | 7.86 | 4.92 | **5.82** |

Table 4.25 WER [%] on the evaluation set of the REVERB challenge simulation data.

| | Room1 | | Room2 | | Room3 | | Avg. |
|---|---|---|---|---|---|---|---|
| | far | near | far | near | far | near | |
| *CE | | | | | | | |
| noisy | 6.44 | 5.76 | 11.91 | 7.46 | 13.27 | 8.21 | 8.84 |
| enhan | 6.44 | 6.05 | 9.89 | 6.12 | 12.04 | 6.21 | 7.79 |
| diff | 6.18 | 5.51 | 9.47 | 6.16 | 11.53 | 7.10 | 7.66 |
| uncert(t) | 6.00 | 5.69 | 9.05 | 5.74 | 11.17 | 6.26 | 7.32 |
| uncert(d) | 6.40 | 5.88 | 9.89 | 6.25 | 12.04 | 6.28 | 7.79 |
| uncert(t,d) | 5.90 | 5.62 | 9.03 | 5.79 | 11.05 | 6.24 | **7.27** |
| *SMBR | | | | | | | |
| noisy | 5.40 | 5.01 | 9.64 | 5.87 | 10.93 | 7.20 | 7.34 |
| enhan | 5.73 | 5.29 | 7.72 | 5.35 | 9.57 | 5.77 | 6.57 |
| diff | 5.37 | 4.95 | 7.83 | 5.45 | 9.67 | 6.19 | 6.58 |
| uncert(t) | 5.40 | 5.13 | 7.81 | 5.58 | 9.31 | 6.12 | 6.56 |
| uncert(d) | 5.45 | 4.98 | 7.89 | 5.51 | 9.40 | 5.83 | 6.51 |
| uncert(t,d) | 5.25 | 5.03 | 7.80 | 5.42 | 9.13 | 5.89 | **6.42** |

proposed method was confirmed for noisy and reverberant two ASR tasks. Future work will seek to develop an algorithm that determines the optimal interpolated points depending on the type of noise.

## 4.9 Conclusion of the chapter

This chapter focused on discriminative training for robust ASR systems. Experimental results showed that discriminative training is effective for feature transformation (Section 4.3.2), acoustic modeling (Sections 4.4.2 and 4.5.2), system combination (Section 4.6), and language modeling

(Section 4.7). To improve the robustness to the speech distortion, uncertainty training and decoding methods of DNN was proposed in Section 4.8.

Journal papers related to this chapter are [254, 255] and conference papers are [256, 206, 219, 257, 258, 259].

# 5 Development of ASR systems for realistic noisy and reverberant environments

## 5.1 Introduction

We have developed various techniques to improve the robustness of ASR under noisy and reverberant environments in previous three chapters. This chapter validates the effectiveness of our proposed methods on realistic noisy ASR tasks. These tasks are open challenge where everyone gets the same data and compares their method in the same criteria. There are four ASR challenges that we participated: the second, third, and fourth CHiME challenge and the REVERB challenge. CHiME series are noisy ASR tasks and REVERB challenge is a reverberant ASR task. The target of the second CHiME challenge is home applications and those of the third and fourth CHiME challenge are tablet ASR in public spaces. REVERB challenge has variety of reverberation, i.e., multiple rooms and multiple speaker-to-microphone distances. In addition, there is one source localization and VAD challenge: DIRHA challenge.

In this chapter, we introduce the details of each challenge and prepared techniques. For each challenge, we prepared some original techniques including above-mentioned methods.

## 5.2 Noisy ASR in house (The second CHiME challenge)

To validate the effectiveness of state-of-the-art speech enhancement and ASR techniques in distant-talking conditions, several challenges have been organized [260, 261, 262]. Among these, the Computational Hearing in Multisource Environments (CHiME) challenges recently introduced noise-robust speech processing tasks with a small number of microphones [263, 264, 260, 261]. The goal of these tasks is to recognize speech from a distant target speaker that was binaurally recorded in a domestic environment. Whereas the first CHiME challenge is a simple keyword recognition task [264, 260], the second CHiME challenge contains a medium vocabulary recognition task (track 2). In particular, track 2 contains simulated speech samples that are taken from the Wall Street Journal (WSJ0) 5k vocabulary read speech corpus, convolved with binaural room impulse responses, and then mixed with binaural recordings of a noisy domestic environment [261]. The second challenge is much more complex and difficult from an ASR point of view. To overcome this challenging task, we propose a system involving state-of-the-

art and newly-proposed components, including a noise suppression method as well as various discriminative training and feature transformation methods.

We propose a BM method based on the estimated TDOA that is to be used for noise suppression; this method takes advantage of the availability of the binaural training data provided by the challenge. If many microphones are available, linear noise suppression techniques are effective and generate little distortion [265]. When only two microphones are used, however, one can expect SNR improvements of up to only 3 dB when using techniques such as standard delay-and-sum BF. Therefore, one needs to resort to non-linear methods for better performance. One such non-linear method is a BM technique based on the TDOA that has been shown to be simple and effective for a small number of microphones [70]. However, the TDOA estimation accuracy can be severely degraded in the presence of reverberation and noise [37]. To compensate for the influence of reverberation and noise, we propose to use the training data to generate a prior distribution of the discrepancy between the instantaneous inter-microphone phase difference and the expected phase difference of sound emanating from the target speaker location. That prior distribution is then used when building the binary mask. We refer to this approach as prior-based BM.

In this section, the goal is not only to improve the baseline ASR systems by using BM approaches, but also to understand to what extent performance can be improved by using the discriminative training ASR approach. While state-of-the-art ASR techniques have been shown to be very effective in clean speech conditions, further investigation is needed in order to improve the effectiveness of ASR techniques in challenging conditions such as in the presence of environmental reverberation and noise. This section proposes an approach to overcome these challenges by evaluating discriminative training and feature transformation techniques based on the samples provided in the second CHiME challenge. As the conditions between the training data and the test data are matched, it is reasonable to expect that discriminative training methods will lead to significant performance improvements even in reverberant and noisy conditions. In particular, we investigate the performance change when using MMI and boosted MMI (bMMI) training. We also investigate previously-mentioned several feature transformation approaches.

Whereas the aforementioned conventional acoustic modeling techniques are mainly used within the GMM framework, this section also investigates their use within the commonly used hybrid DNN-HMM approach [137]. The study includes all of the previously mentioned discriminative training and linear feature transformation techniques, but excludes f-MMI/f-bMMI[1]. The experimental evaluation shows that these techniques still continue to effectively improve performance when used with a DNN.

In the ASR post-processing step, we propose to use a re-scoring technique based on a simple combination of DLM (Section 4.7.1) and MBR decoding [266, 267, 268, 269]. In contrast with [269], which performs DLM with the MBR criterion, our work combines DLM and MBR *decoding* in a cascade form; we simply use the re-ranked 1-best obtained through DLM to initial-

[1]This is a reasonable exclusion because the lower layers of the DNNs already serve as an effective non-linear feature transformation.

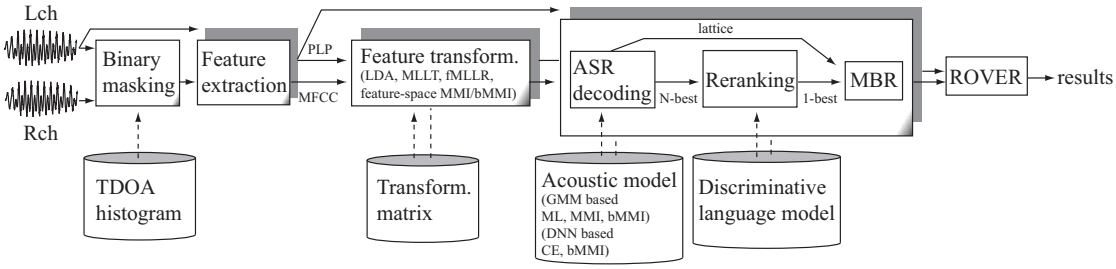



Fig. 5.1 Schematic diagram of the proposed system.

ize the MBR decoding. As a final step, system combination e.g., recognizer output voting error reduction (ROVER) [195] and its variants [197, 198, 206] can be used to obtain refined hypotheses by majority voting of the hypotheses of different systems; this results in higher performance than each base system can achieve individually. In order to create systems with complementary hypotheses, this work constructs two systems based on MFCC features as well as PLP features [270].

In summary, the goal of this section is to evaluate the effectiveness of various state-of-the-art and novel techniques for ASR in reverberant and noisy environments. In particular, the techniques providing additional novel approaches are the prior-based BM (Section 5.2.2) and the combination of DLM and MBR (Section 5.2.4).

### 5.2.1 System overview

Fig. 5.1 is a schematic diagram of the proposed system, which consists of three components. First is the noise suppression step, which is a prior-based BM that suppresses directional interferences (Section 5.2.2). Second is the feature transformation step, including feature-level transformations (LDA and MLLT with/without fMLLR, which are conventional and thus not explained in detail here) as well as discriminative feature transformations (feature-space techniques, presented in Section 4.4.6). Third is the ASR decoding step; it uses an acoustic model (GMM/DNN) with sequence discriminative training (Sections 4.4.2 and 4.4.4). Decoding results are re-ranked using DLM (Section 4.7.1), and MBR is performed based on the DLM output (Section 5.2.3). The best results were obtained by the ROVER combination of the hypotheses of two DNN systems using different features (MFCC and PLP).

### 5.2.2 Prior-based binary masking

In the CHiME challenge, two-channel recordings are provided and the target speaker is in a fixed frontal position with respect to the microphones[2]. BM based on the TDOA has been

[2]This is a reasonable setting suitable for many applications, in which the users are either in a frontal position (such as when using home appliances), or in a fixed position (such as when using car navigation systems). In

shown to be more effective when used for ASR with a small number of microphones than simple delay-and-sum BF [70]. Consequently, we investigate the usage of this BM technique in our system.

When the receiver is in a frontal position and there is little reverberation and noise, the TDOA for signals coming from the target speaker should be close to zero. Hence, time-frequency bins for which the inter-microphone phase difference is not close to zero are unlikely to contain energy from the target speaker. However, in the presence of reverberation, the phase differences of the sound waves from a frontal target source may be non-zero. Fig. 5.2 shows the phase difference histograms at 250 Hz and 1 kHz in the "reverberated" (i.e., no noise) speech of the CHiME challenge training set. At 250 Hz, the histogram is almost symmetrical and the variance is small; at 1 kHz, however, the mean has drifted and the variance is large. The extent to which the phase difference is affected by reverberation and noise varies significantly for each frequency bin. Thus, a simple binary mask using only physical information will not be effective; indeed, preliminary experiments showed that this type of binary mask led to a word error rate (WER) that was worse than the baseline. As in [26], a statistical model is needed. In order to account for the offset of the phase difference when compared to the anechoic case as well as its variance, a prior-based BM is proposed. The phase difference $\theta_{t,\omega}$ at time frame $t$ and frequency bin $\omega$ is calculated for each time-frequency bin as

$$\theta_{t,\omega} = \angle \left( X^L_{t,\omega} / X^R_{t,\omega} \right) \in (-\pi, \pi], \tag{5.1}$$

where $X^L_{t,\omega}$ and $X^R_{t,\omega}$ are the complex short-time Fourier spectra for the left and right channels, respectively, and $\angle$ denotes the argument operator of a complex number.

In classical BM, a time-varying masking vector $\boldsymbol{W}_t = [W_{t,1}, \ldots, W_{t,\omega}, \ldots, W_{t,\Omega}]^\top \in \mathbb{R}^\Omega$ (where $\top$ denotes transposition) is designed using the following thresholding function:

$$W_{t,\omega} = \begin{cases} \epsilon & (|\theta_{t,\omega}| > \theta_c), \\ 1 & (\text{otherwise}), \end{cases} \tag{5.2}$$

where $\epsilon$ is a very small constant for spectral smoothing, and $\theta_c$ is a threshold determined in advance. Noise suppressed spectra $\boldsymbol{Y}_t \in \mathbb{C}^\Omega$ are obtained as

$$\boldsymbol{Y}_t = \boldsymbol{W}_t \odot \left( \boldsymbol{X}^L_t + \boldsymbol{X}^R_t \right) / 2, \tag{5.3}$$

where $\boldsymbol{X}^L_t, \boldsymbol{X}^R_t \in \mathbb{C}^\Omega$, and $\odot$ denotes the element-wise multiplication of two vectors.

In our prior-based BM approach, a time-varying masking vector $\boldsymbol{W}'_t$ is determined using a frequency-dependent prior probability $q_\omega(\theta)$ of the phase difference $\theta$. This prior probability is obtained from a phase difference histogram computed on the training data, renormalized to sum to unity. Denoting the peak of the histogram for frequency $\omega$ as $\bar{q}_\omega = \max_\theta q_\omega(\theta)$, we define the masking vector as

$$W'_{t,\omega} = \begin{cases} \epsilon & (q_\omega(\theta_{t,\omega}) / \bar{q}_\omega < q_c), \\ \left( q_\omega(\theta_{t,\omega}) / \bar{q}_\omega \right)^\alpha & (\text{otherwise}), \end{cases} \tag{5.4}$$

a situation where speakers are able to move freely, our prior-based BM approach could be modified to allow for multiple priors according to the speaker direction; this direction could be estimated by another method such as the cross-spectrum phase method [29].

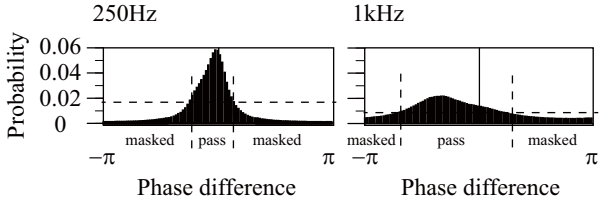



Fig. 5.2 Histogram of phase differences for two frequency bins.

where $q_c$ is a threshold that determines the relative height with respect to the peak above which a time-frequency bin is passed. $\alpha$ is a warping parameter that can set the behavior of the mask from soft to binary. Both $q_c$ and $\alpha$ are tuned manually in the development set. Whereas in classical BM, thresholding is based on a constant tolerance angle between the reference and the observation, our thresholding function takes the shape of the histogram into account. For histograms with a pronounced peak, such as the one corresponding to the 250 Hz frequency bin in Fig. 5.2, the tolerance angle is small, and only time-frequency bins for which the phase difference is very close to the peak are passed by the mask. On the other hand, the tolerance angle is large for flatter histograms such as the one corresponding to the 1 kHz bin in Fig. 5.2; in the latter case, phase differences farther from the peak are passed as well.

### 5.2.3 Minimum Bayes risk decoding

MBR decoding is another re-scoring technique that attempts to approximately minimize the Bayes risk obtained from the WER [266, 267, 268]. The algorithm modifies the 1-best word sequence $s_1^{(r)}$ by word-by-word replacements to obtain a modified word sequence $\tilde{s}^{(r)}$ that minimizes the expected edit distance $L(\tilde{s}^{(r)}, s')$ to other word sequences $s'$ in the hypothesis lattice[3] $\mathcal{L}^{(r)}$. The edit distance $L$ is approximately computed based on the forward-backward algorithm [268] and this procedure repeats until no symbols are replaced.

### 5.2.4 Combination of minimum Bayes risk decoding with discriminative language modeling

In previous section, conventional MBR decoding starts from the 1-best word sequence of the lattice and then forming alignments of the rest of hypotheses. The iteration above can reach local minimum, similar to the ML training in acoustic modeling. Our approach improves the initial point by replacing the conventional 1-best word sequence $s_1^{(r)}$ with the 1-best word sequence $\hat{s}$ in an $N$-best list re-scored by DLM (Section 4.7.1) [4] to efficiently combine minimum Bayes risk

[3] $N$-best lists can be used instead of lattices.

[4] The accurate assignment probability can be obtained by converting the estimated DLM weights to arc weights in a lattice. However, the conversion is not trivial since DLM would include unseen n-gram features or wide-span features, and the corresponding DLM weights cannot be converted to those of lattice arcs, in a straightforward manner.

Table 5.1 Number of utterances and speakers in each dataset of the second CHiME challenge. Development and evaluation datasets were provided for each SNR.

| dataset | # utterances | # speakers |
|---|---|---|
| Training dataset (**si_tr_s**) | 7,138 | 83 |
| Development dataset (**si_dt_05**) | 409 | 10 |
| Evaluation dataset (**si_et_05**) | 330 | 12 |

decoding with DLM-based $N$-best re-scoring.

### 5.2.5 System combination

A combination of multiple systems, even if some of the systems have significantly lower performance, may outperform the best single system, in particular when the systems tend to display different patterns in their errors. Many system combination methods, such as [195, 197, 198, 271, 272], have been proposed. Here, we use ROVER [195], which is the simplest approach, because system combination is complementary component of this section. ROVER combines the 1-best results outputs of multiple systems which mainly differ by their input features, MFCC and PLP.

### 5.2.6 Experimental setups

The track 2 of the second CHiME challenge [261] is a medium-vocabulary task whose speech utterances are taken from the *Wall Street Journal* database (WSJ0). Table 5.1 presents detailed information about the training (**si_tr_s**), development (**si_dt_05**), and evaluation (**si_et_05**) datasets. Table 5.2 shows the settings for the ASR systems.

Acoustic models were trained using the **si_tr_s** and some of the parameters (e.g., language model weights) were tuned using the WERs on the **si_dt_05**. This database simulates realistic environments. There are two types of data, "reverberated" and "isolated". The "reverberated" data were created by convolving clean speech with binaural room impulse responses corresponding to a frontal position at a distance of 2 m from the stereo microphones in a family living room. The "isolated" data were created by adding real-world noises recorded in the same room to the "reverberated" data, and then adding noise excerpts selected to obtain signal-to-noise ratio (SNR) ranges of $-6$, $-3$, 0, 3, 6, and 9 dB without rescaling. Added noise sources are typically non-stationary (e.g., other speakers' utterances, home noises, or music). We used Kaldi toolkit [124] for the experiments.

#### 5.2.6.1 Feature extraction and transformation

We now describe the settings of the feature extraction and the feature transformation. The baseline acoustic features were MFCCs. In addition to these, PLP features were used for the final system combination, as described in Section 5.2.5. In this section, the LDA classes are

Table 5.2 Setup for the ASR systems.

| | |
|---|---|
| Sampling frequency | 16 kHz |
| Window length | 25 ms |
| Window shift | 10 ms |
| Feature 1 | 0th∼12th MFCCs/PLPs + Δ + ΔΔ |
| Feature 2 | (0th∼12th MFCCs/PLPs × 9 frames) + LDA+MLLT (→40 dim.) |
| Feature 3 | 0th∼22th filter banks (FBANK) + Δ + ΔΔ |
| HMM state | 2,500 shared triphone states |
| Number of Gaussians | 15,000 |
| Hidden layer of DNN | 3 |
| Vocabulary size | 5,000 |

taken as the tri-phone HMM states. We concatenate 13-order static MFCCs in nine contiguous frames to consider the influence of long context, instead of using conventional delta features. This results in a total of 117-dimensional features, which are compressed into 40 dimensions. We use diagonal-covariance models, together with MLLT feature space transformation to decrease correlations between features.

For DNN, mel filter bank (FBANK) features tend to lead to better performance than MFCC features. We validate the effectiveness of FBANK features in addition to MFCC features and MFCC + LDA+MLLT features. For further noise robustness, we also investigate the use of SAT and global fMLLR.

#### 5.2.6.2 Discriminative feature transformation

In discriminative feature transformation (Section 4.4.6), the UBM is constructed using $N_g(= 400)$ Gaussians. Offset features are calculated for each of $K_g(= 39)$-dimensional MFCC features including Δ and ΔΔs, and the posterior probabilities are expanded using nine contiguous frames. The total dimension of the feature vector $\boldsymbol{h}_t$ is 144k (400 [Gaussians] × (39 + 1) [dimensions/Gaussian/frame] × 9 [frames]). Features with the top two posteriors are selected and all other features are set to zero.

#### 5.2.6.3 Acoustic models

We summarize the experimental procedure based on the above setup as follows: First, a clean acoustic model was trained. The number of mono-phones was 40, including silence ("sil"). Second, reverberated acoustic models were trained using the "reverberated" dataset. Third, noisy acoustic models were trained multi-conditionally using the "isolated" dataset without noise suppression. Finally, from this ML model, the effectiveness of the discriminative training and feature transformation for the "isolated" dataset was validated. The parameters used in our experiments were set to be those described in the WSJ tutorial attached to the Kaldi toolkit.

For the DNN, we used the nnet2 of neural network training implemented in the Kaldi toolkit with three hidden layers whose activation functions were sigmoid. Stacking hidden layers layer by layer, the DNN was constructed instead of using the restricted Boltzmann machine. The learning rate $\eta$ was decreased from the initial learning rate $\eta_0$ (0.01) to the final learning rate $\eta_e$ (0.001) at the end of training as $\eta = \eta_0 \exp(i \ln(\eta_e/\eta_0)/i_{\max}))$ where $i$ is an iteration number. The number of iterations $i_{\max}$ was 43 and the minibatch size was 128. Nine concatenated frames were input and the number of hidden layer nodes was 309.

#### 5.2.6.4 Discriminative language modeling

Weights $\boldsymbol{w}$ in Eq. (4.65) of a DLM were learned on the training data set using 100-best recognition candidates, where the weight $w_0$ associated with the original score was set to 20. Using these weights, results were re-ranked, with $w_0$ set to 13. Weights were obtained by averaged perceptron at three iterations. Features were counts of uni-, bi-, and tri-grams.

#### 5.2.6.5 System combination

System combination techniques are effective for the case in which the hypotheses of the respective systems are different but the performance of the systems is similar. The most promising approach is to use additional features; thus, after generation of the best hypotheses of the DNN-HMM system for MFCC and PLP feature with regard to the time alignment and the confidence measure, these hypotheses were combined using ROVER.

### 5.2.7 Results and discussion

#### 5.2.7.1 Discriminative training

With regard to the MFCC features, discriminative training improved the WER from the ML baseline as shown in Table 5.3 (upper)[5]. The mixture of speech and noise increases the likelihood of detecting erroneous phonemes and leads to incorrect recognition especially when the noise source is other people's utterances. These errors could be modified by discriminative training. The boosting factor in Eqs. (4.38), (4.43), and (4.50), $b$, was set to 0.1 because the preliminary experiments show that the performance did not heavily depend on the boosting factors and that the optimized values of the boosting factor were approximately 0.1–0.2. The denominator lattices for discriminative training were generated using the ML model. The boosted MMI improved the WER by 1.6% absolute[6] to the ML, whereas the feature-space discriminative training improved the WER by 3% further. We believe that the feature space was adapted for a target speaker to improve the WER and that this effect reduced the influence of other noises.

[5]The MMI and f-MMI results were omitted, because the performance of those was lower than those of the bMMI and f-bMMI and recently, the results of GMM were less meaningful than at the time of the second CHiME challenge. The detailed evaluations are found in [183].

[6]In this section, WER improvements are shown in absolute values

Table 5.3 WER[%] of GMM-HMM for **si_dt_05** without noise suppression. MFCC features (upper), MFCC + LDA+MLLT (middle), MFCC + LDA+MLLT + SAT+fMLLR (lower).

∘MFCC + Δ + ΔΔ

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| ML | 74.20 | 66.57 | 58.24 | 51.84 | 46.73 | 40.64 | 56.37 |
| bMMI | 72.78 | 64.71 | 55.69 | 50.83 | 44.00 | 40.27 | 54.71 |
| f-bMMI | 68.64 | 61.56 | 53.11 | 47.65 | 41.73 | 36.98 | 51.61 |

∘MFCC + LDA+MLLT

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ML | 70.95 | 62.62 | 53.98 | 47.37 | 40.27 | 34.84 | 51.67 |
| f-bMMI | 66.65 | 57.46 | 48.25 | 42.99 | 35.71 | 31.07 | 47.02 |

∘MFCC + LDA+MLLT + SAT+fMLLR

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ML | 68.36 | 58.30 | 48.80 | 40.73 | 35.09 | 28.54 | 46.64 |
| f-bMMI | **62.43** | **52.23** | **42.17** | **35.31** | **29.84** | **24.72** | **41.12** |

#### 5.2.7.2 Feature transformation

The MFCC features were transformed using LDA and MLLT. Table 5.3 (middle) shows the WER for this case, whereas LDA by itself (i.e., without MLLT) achieves 54.37% (ML). This shows that features that are highly discriminable from other phonemes can be obtained by LDA. The performance gains of LDA and MLLT were 2.0 and 2.7%, respectively. It is effective to use a long context to reduce the influence of non-stationary noises. Furthermore, although noises increase the correlations between MFCC coefficients in each dimension, MLLT reduced the correlations. The denominator lattices for discriminative training were re-generated using the ML (MFCC + LDA+MLLT) model. Discriminative training improved the WER by 4.6%.

#### 5.2.7.3 Adaptation

Table 5.3 (lower) shows the WER when additional SAT and fMLLR were used. Because the amount of training data is very limited, transformation into a canonical space, which leads to an increase in the effective amount of training data, has a strong impact on the estimation accuracy of the acoustic models. Additionally, fMLLR adaptation for a target speaker reduced the influence of noises and improved the WER by 5.0%. The denominator lattices for discriminative training were also re-generated using this adapted ML model. Discriminative training improved the WER by 5.5%.

#### 5.2.7.4 Noise suppression

In order to clarify the effectiveness of the prior-based proposed BM, Table 5.4 shows the WERs of the proposed BM compared with those of the conventional BM [70] by using baseline GMM with MFCC features. As mentioned in the section 5.2.2, the conventional BM improved the performance significantly, whereas the proposed BM improved the WER in all SNRs by 7% to

Table 5.4 WER[%] of GMM-HMM for **si_dt_05** with noise suppression by conventional binary masking (BM) and the proposed prior-based BM. MFCC features were used.

∘MFCC + Δ + ΔΔ

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| conventional BM | 73.98 | 66.90 | 57.93 | 52.35 | 46.38 | 40.54 | 56.35 |
| prior based BM | 66.82 | 57.87 | 48.86 | 42.29 | 38.18 | 31.86 | 47.65 |

Table 5.5 WER[%] of GMM-HMM for **si_dt_05** with noise suppression by prior-based BM. MFCC features (upper) and MFCC + LDA+MLLT + SAT+fMLLR (lower).

∘MFCC + Δ + ΔΔ

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| f-bMMI | 63.40 | 54.05 | 44.28 | 38.87 | 33.72 | 29.90 | 44.04 |
| ∘MFCC + LDA+MLLT + SAT+fMLLR | | | | | | | |
| ML | 59.94 | 47.93 | 39.83 | 33.01 | 28.00 | 23.47 | 38.70 |
| f-bMMI | 52.93 | 42.62 | 34.59 | 27.63 | 24.27 | 20.24 | 33.71 |
| (+DLM) | 53.16 | 42.93 | 34.36 | 27.26 | 23.72 | 19.47 | 33.48 |
| (+MBR) | 52.65 | 42.04 | 33.75 | 27.05 | 23.74 | 19.91 | 33.19 |
| (+DLM+MBR) | 52.54 | 42.09 | 33.72 | 27.02 | 23.66 | 19.66 | 33.11 |

Table 5.6 WER[%] of DNN-HMM for **si_dt_05** without noise suppression. MFCC features (upper) and MFCC + LDA+MLLT (lower).

∘MFCC + Δ + ΔΔ

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| CE | 67.47 | 57.55 | 48.78 | 43.43 | 36.10 | 31.76 | 47.52 |
| ∘MFCC + LDA+MLLT | | | | | | | |
| CE | 64.39 | 53.67 | 44.28 | 38.56 | 32.70 | 28.09 | 43.62 |

9%. The best warping parameters for the proposed BM $\alpha$ was 0.25. Directional noises were effectively suppressed by our proposed method, but diffused noises such as music remained.

Table 5.5 shows the WER with feature adaptation and discriminative training. Combination of them with noise suppression was effective. The employed adaptation improved the WER by 9.0% and discriminative training improved it by 5.6%.

#### 5.2.7.5 Deep neural network

Table 5.6 and 5.7 provide the WERs of a DNN. Table 5.6 shows the result without noise suppression and Table 5.7 shows that with noise suppression. Using the same MFCC features, at the ML and CE baseline, the DNN result outperformed the GMM results by 8.9% (without noise suppression) and 6.3% (without noise suppression), respectively.

Table 5.7 (the second division) shows that the FBANK features outperformed the MFCC features, as previous studies have shown. The performance of MFCC + LDA+MLLT was worse

Table 5.7 WER[%] of DNN-HMM for **si_dt_05** with noise suppression by prior-based BM. MFCC features (first), FBANK features (second), MFCC + LDA+MLLT (third), MFCC + LDA+MLLT + SAT+fMLLR (fourth) and PLP + LDA+MLLT + SAT+fMLLR (fifth). Hypotheses of two systems (*1 and *2) were combined by ROVER (last).

| ∘MFCC + Δ + ΔΔ | | | | | | | |
|---|---|---|---|---|---|---|---|
| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
| CE | 62.44 | 51.59 | 42.93 | 35.27 | 30.11 | 25.76 | 41.35 |
| ∘FBANK + Δ + ΔΔ | | | | | | | |
| CE | 55.60 | 44.52 | 36.16 | 30.62 | 26.02 | 22.32 | 35.87 |
| bMMI | 51.70 | 39.43 | 31.75 | 26.83 | 23.35 | 19.86 | 32.15 |
| ∘MFCC + LDA+MLLT | | | | | | | |
| CE | 57.21 | 45.85 | 36.21 | 30.61 | 26.36 | 23.31 | 36.59 |
| ∘MFCC + LDA+MLLT + SAT+fMLLR | | | | | | | |
| CE | 52.78 | 42.50 | 34.08 | 27.05 | 24.13 | 20.12 | 33.44 |
| bMMI | 47.34 | 36.33 | 28.96 | 23.40 | 20.03 | 17.05 | 28.85 |
| (+DLM) | 47.37 | 36.48 | 28.94 | 23.09 | 20.02 | 16.93 | 28.80 |
| (+MBR) | 46.79 | 35.68 | 28.44 | 22.88 | 19.91 | 16.64 | 28.39 |
| *1 (+DLM+MBR) | 46.67 | 35.55 | 28.38 | 22.84 | 19.83 | 16.65 | 28.32 |
| ∘PLP + LDA+MLLT + SAT+fMLLR | | | | | | | |
| *2 (+DLM+MBR) | 47.38 | 35.29 | 27.89 | 22.70 | 19.38 | 15.92 | 28.09 |
| ∘ROVER | | | | | | | |
| *1+*2 | **45.12** | **34.34** | **26.73** | **21.71** | **19.09** | **15.39** | **27.06** |

than that of the FBANK features for a DNN-HMM system. When combined with GMM-based speaker adaptation techniques (SAT+fMLLR), DNN slightly outperformed f-bMMI even without discriminative training when Table 5.7 is compared with Table 5.5[7]. With discriminative training (bMMI) for DNN, the DNN outperformed f-bMMI of GMM by 4.9%. This shows the effectiveness of DNN for noise-robust ASR. The performance gains by discriminative training of acoustic models were around 5% for both GMM and DNN.

#### 5.2.7.6 Discriminative language modeling and minimum Bayes risk decoding

Table 5.5 (lower) and 5.7 (the fourth division) show that DLM improved the average WER by 0.2% and 0.05%, respectively, especially for the 9dB case of GMM, which resulted in a 0.8% improvement. DLM was not always effective because, while error tendencies were dependent on a particular SNR, training was performed on the whole multi-condition training set, which included all SNRs. This led to a mismatch between training and recognition, thereby degrading performance. DLM was less effective for DNN than GMM.

[7]This type of adaptation cannot be directly applied for the FBANK feature due to their high dimensionality and correlation across feature dimensions [160].

Table 5.8 WER[%] of GMM-HMM for **si_et_05** without noise suppression.

| ∘MFCC + Δ + ΔΔ | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| -GMM-HMM | | | | | | | |
| ML | 69.79 | 62.71 | 55.86 | 46.89 | 42.07 | 37.49 | 52.47 |
| -DNN-HMM | | | | | | | |
| CE | 62.79 | 53.19 | 46.46 | 38.26 | 32.30 | 30.34 | 43.89 |
| ∘MFCC + LDA+MLLT + SAT+fMLLR | | | | | | | |
| -GMM-HMM | | | | | | | |
| ML | 60.83 | 52.14 | 43.51 | 34.28 | 29.22 | 23.82 | 40.63 |
| f-bMMI | **54.70** | **45.11** | **35.98** | **28.64** | **24.38** | **21.39** | **35.04** |

#### 5.2.7.7 Minimum Bayes risk decoding and combination with discriminative language modeling

Table 5.5 (lower) and 5.7 (the fourth division) show that MBR improved the WER by 0.5% for both GMM and DNN. The performance of MBR was stable with respect to SNR. The combination of DLM and MBR as mentioned in Section 5.2.3 improved the WER further by 0.1% for both cases because DLM refined the initial 1-best result and adapts to error tendencies inherent to the decoder. Thus, MBR was effective for both GMM and DNN.

#### 5.2.7.8 System combination

Table 5.7 (the fifth division) shows the WER using PLP features for the best case of DNN. This (PLP) result was equivalent to the condition of 1) of the fourth division. PLP was slightly better than MFCC but preliminary experiments show that simple concatenation of MFCC and PLP features for DNN degraded the performance. Table 5.7 (the last division) shows that ROVER, which combined the 1-best hypotheses of MFCC and PLP, improved the WER by 1% and this was effective in all SNR cases.

#### 5.2.7.9 Evaluation set

Table 5.8 shows the WERs on the evaluation set using the models tuned on the development set. Tendencies were the same to those of the development set. DNN was still effective for evaluation set. Using both discriminative training and feature transformation (f-bMMI) achieved a 33.2% error reduction relative to the baseline (ML). Thus, we show the effectiveness of both discriminative training and feature transformation for reverberated and noisy speech.

Table 5.9 shows the WERs after noise suppression. Using a GMM-HMM system with both discriminative training and feature transformation (f-bMMI) achieved a 37.9% error reduction relative to the baseline (ML). These results were submitted to the CHiME challenge workshop [183]. Moreover, for this case, DNN with bMMI and system combination of two systems

Table 5.9 WER[%] of GMM- and DNN-HMM for **si_et_05** with noise suppression.

| | −6dB | −3dB | 0dB | 3dB | 6dB | 9dB | Avg. |
|---|---|---|---|---|---|---|---|
| ∘MFCC+ Δ + ΔΔ | | | | | | | |
| -GMM-HMM | | | | | | | |
| ML | 60.58 | 52.87 | 45.60 | 37.70 | 33.38 | 29.24 | 43.23 |
| ∘MFCC + LDA+MLLT + SAT+fMLLR | | | | | | | |
| -GMM-HMM | | | | | | | |
| ML | 50.91 | 41.64 | 33.89 | 26.30 | 21.61 | 18.85 | 32.20 |
| f-bMMI | 44.54 | 35.91 | 29.24 | 22.31 | 17.77 | 15.88 | 27.61 |
| (+DLM) | 44.27 | 35.48 | 28.75 | 21.61 | 17.34 | 15.37 | 27.14 |
| (+MBR) | 44.51 | 35.42 | 28.81 | 21.46 | 17.41 | 14.98 | 27.10 |
| (+DLM+MBR) | 44.12 | 35.46 | 28.12 | 21.20 | 17.43 | 14.83 | 26.86 |
| -DNN-HMM | | | | | | | |
| bMMI | 37.98 | 28.26 | 21.86 | 17.71 | 12.61 | 11.75 | 21.70 |
| (+DLM) | 38.00 | 27.82 | 21.80 | 16.64 | 12.22 | 11.62 | 21.35 |
| (+MBR) | 37.14 | 27.35 | 21.41 | 16.94 | 12.55 | 11.54 | 21.16 |
| *1 (+DLM+MBR) | 37.16 | 27.44 | 21.24 | 16.66 | 12.40 | 11.49 | 21.07 |
| ∘PLP + LDA+MLLT + SAT+fMLLR | | | | | | | |
| -DNN-HMM | | | | | | | |
| *2 (+DLM+MBR) | 38.22 | 27.93 | 22.57 | 16.91 | 13.49 | 12.14 | 21.88 |
| ∘ROVER | | | | | | | |
| *1+*2 | **36.43** | **26.02** | **20.96** | **15.84** | **11.99** | **11.17** | **20.40** |

(ROVER) achieved a 52.6% error reduction, which means that errors were reduced by more than half.

### 5.2.8 Conclusion

We developed a state-of-the-art recognition system for the second CHiME challenge track 2, which is a medium-size automatic speech recognition task under noisy environments, and validated the effectiveness of both feature transformation and discriminative methods. For realistic reverberated and noisy environments of this task, we proposed a prior-based binary masking and show its effectiveness. Combination of minimum Bayes risk decoding and discriminative language modeling improved the word error rate by considering error tendencies, which are inherent to the decoder. Deep neural networks are also effective; they outperformed the feature-space boosted maximum mutual information technique, which had been the state-of-the-art acoustic modeling technique for conventional Gaussian mixture model based systems. This superior performance was achieved even without discriminative training; with the combination of sequential discriminative training and system combination, the best performance was achieved. Experiments show that these techniques are effective for non-stationary interference and reverberation.

Future work will be an extension of our approaches to various tasks. For handling distant speech, reverberation effect is also important [262]. In this scenario, because the speaker moves

freely, our prior-based binary masking approach needs modifications to include multiple priors according to the speaker direction.

## 5.3 Noisy ASR in public spaces 1 (The third CHiME challenge)

The aim of this challenge is the same to that of the second CHiME challenge. Many methods have been proposed to improve ASR performance under noisy environments such as SE, feature transformation, and discriminative methods. Each method has a specialty for specific noise and no universal solution exists. The optimal ASR system is different for each utterance and the number of their combination is enormous. Although overall combination of their hypotheses can improve the performance, the computational resources increase in proportion to the number of systems. If the single optimal system can be picked up from many ASR systems prior to SE and ASR decoding, the computational resources do not increase. For example, if there are two systems and the first system is apparently superior to the second system for environment A and the second system is superior to the first system for environment B, it is better to select an optimal single system than to combine systems in terms of a computational resource.

This section proposes an efficient system selection method based on the estimated WERs of ASR systems before performing SE and ASR decoding. Previous studies [112, 113] used perceptual evaluation speech quality (PESQ) scores to predict WERs; however, the calculation of PESQ scores requires clean speech, which cannot be obtained for evaluation data. Even if PESQ scores can be obtained, these kinds of estimations also require enhanced speech, thus it is necessary to perform at least SE before selection. On the other hand, limited to reverberation, the performance is estimated from room acoustic parameters [113, 273] but these types of estimations need room acoustic impulse responses and this is not a realistic assumption. Another study [274] used recognition hypotheses; however, in order to select the optimal SE method, it is inefficient to perform SE and ASR decoding for every system. In addition, if multiple hypotheses have been already obtained, system combination is better than system selection. Our method uses i-vectors, which represent speaker and channel characteristics [275, 276] of original noisy speech for estimating WERs via cosine similarities between the training and test data. It is unnecessary to perform not only ASR decoding but also SE. A related approach is [277], which uses i-vectors for clustering training data but whose objective is different from our approach.

This section validates the effectiveness of the proposed approach on the third CHiME challenge [278]. The third one is also a medium vocabulary task, which aims to improve the performance of ASR systems in four different public environments such as cafés or streets by using six tablet-embedded microphones. In addition, there are two different conditions in the third CHiME challenge: real ("Real") and simulation data ("Sim."). To overcome this challenging task, we prepare multiple ASR systems with different SE methods and various feature transformations. As mentioned above, the optimal system is different for each environment. In this case, the SE method attached to the challenge baseline performs well for Sim., whereas our employed SE method (maximum SNR-BF [279]) performs well for Real. For this type of situation, system combinations–e.g., recognizer output voting error reduction (ROVER) [195]–can refine hypotheses by majority voting of the hypotheses of multiple systems. Actually, when increased computa-

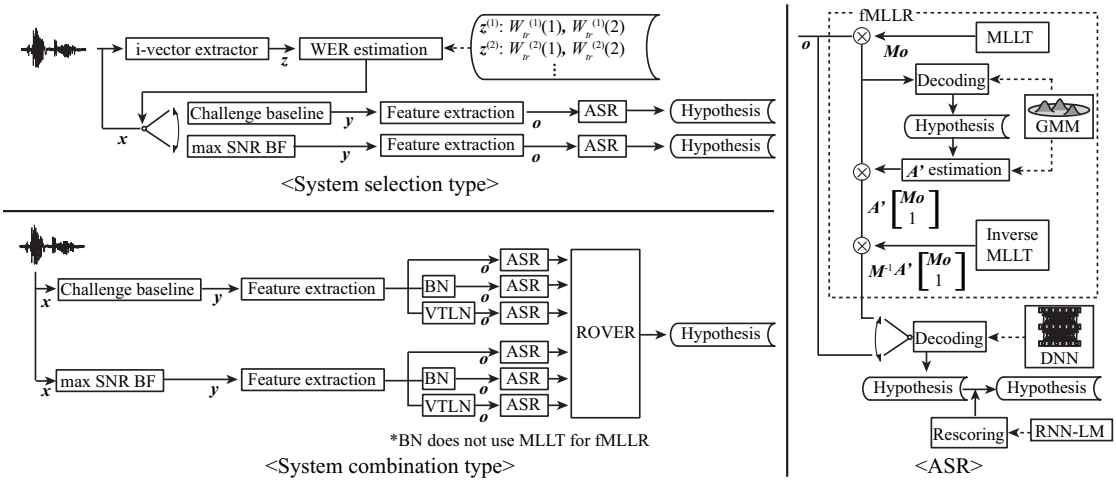



Fig. 5.3 Schematic diagram of the proposed ASR systems.

tional resources can be ignored, system combination is a more robust solution for mismatch and diversity of environments [280, 281]. Experiments show that the proposed optimal ASR system selection method is effective to exploit the better performance from multiple different systems without increasing computational resources.

### 5.3.1 System overview

Fig. 5.3 shows two types of systems; one is the proposed system selection type and the other is a conventional system combination type. The system selection type selects a single system based on i-vectors (Section 5.3.3) whereas the system combination type combines multiple systems' hypotheses to refine the hypotheses by ROVER. There are multiple systems using different SE methods and different feature transformations. Each system has a noise suppression component (CHiME challenge-provided baseline and max SNR BF (Section 5.3.2)) and an ASR decoding component. The ASR decoding component uses either Gaussian mixture model (GMM) or deep neural network (DNN) acoustic model with sequence discriminative training (Section 4.4.5) after feature transformation including bottleneck (BN) features and vocal tract length normalizations (VTLNs) and feature adaptation (Section 4.2.3.2 and 4.2.3.4). In addition, rescoring of language model scores is used by an interpolation of original tri-gram model scores and recurrent neural network language model (RNN-LM) scores.

### 5.3.2 Speech enhancement

SE is performed before ASR and a blind SE method is used because speaker positions are unstable. Two types of blind methods are prepared.

#### 5.3.2.1 Challenge baseline

This method estimates a direction of arrival by a nonlinear SRP-PHAT pseudo-spectrum [282]. After target direction is obtained, Viterbi algorithm is used for calculating transition probabilities between successive speaker positions. These probabilities are related to the distance between the speaker and microphone array. The multichannel spatial covariance matrices are estimated from noise signals in 5 seconds, which are added before the speech. Using these matrices, time-varying minimum variance distortionless response beamforming with diagonal loading [283] enhances speech with taking possible microphone failures into account.

#### 5.3.2.2 Maximum SNR BF

In addition to the challenge baseline, we employ a maximum signal-to-noise ratio (max SNR) beamformer (BF) [279], which is one of the statistically optimal BFs [284]. The enhanced speech spectrum at frame $t$ and frequency bin $\omega$, $y_{t,\omega} \in \mathbb{C}$, is obtained from $N_c$ch original spectrum $\boldsymbol{x}_{t,\omega} \in \mathbb{C}^{N_c \times 1}$ with a mask $\boldsymbol{w}_\omega \in \mathbb{C}^{1 \times N_c}$:

$$y_{t,\omega} = \boldsymbol{w}_\omega \boldsymbol{x}_{t,\omega}. \tag{5.5}$$

According to the voice activity detection results, SNR $\lambda_\omega$ is defined as

$$\lambda_\omega = \frac{\boldsymbol{w}_\omega \boldsymbol{R}_s \boldsymbol{w}_\omega^{\mathrm{H}}}{\boldsymbol{w}_\omega \boldsymbol{R}_n \boldsymbol{w}_\omega^{\mathrm{H}}}, \tag{5.6}$$

where $\boldsymbol{R}_s$ and $\boldsymbol{R}_n$ are covariance matrices in the speech and noise frames, respectively, and H denotes the Hermitian transpose operation. The mask $\boldsymbol{w}_\omega$ that maximizes SNR $\lambda_\omega$ corresponds to a solution to a general eigenvalue problem:

$$\boldsymbol{w}_\omega \boldsymbol{R}_s^{\mathrm{H}} = \lambda_\omega \boldsymbol{w}_\omega \boldsymbol{R}_n^{\mathrm{H}}. \tag{5.7}$$

### 5.3.3 Optimal ASR system selection based on an estimated WER via i-vector similarities

We propose an efficient optimal system selection method that estimates the best performing single system among multiple systems for an unknown utterance based on the i-vector [275, 276]. For all training data, WERs per utterance, $W_{tr}$, are obtained a priori.

i-vectors are derived from a factor analysis that decomposes speech into a speaker/channel invariant part and a variant part as

$$\boldsymbol{V}^{(r)} = \boldsymbol{v} + \boldsymbol{T}\boldsymbol{z}^{(r)}, \tag{5.8}$$

where $\boldsymbol{V}^{(r)}$ is a GMM super vector adapted to the utterance $r$ and is dependent on a speaker and a channel; $\boldsymbol{v}$ is a GMM super vector, which is independent of the speaker and the channel and

**Algorithm 4** Algorithm of the proposed optimal system selection method

**Input:** i-vector for all training data $\boldsymbol{z}_{tr}$, and WER for all training data and all prepared ASR systems $W_{tr}(i)$ where $i$ is a system ID

**for** $r_{ev} = 1$ to (# of evaluation utterances) **do**

Extract i-vector $\boldsymbol{z}_{ev}^{(r_{ev})}$

**for** $r_{tr} = 1$ to (# of training utterances) **do**

Compute similarities $\sigma\left(\boldsymbol{z}_{ev}^{(r_{ev})}, \boldsymbol{z}_{tr}^{(r_{tr})}\right)$

**end for**

Find the most similar utterance $\hat{r}_{tr}$ as in Eq. (5.9)

Find the best ASR system $\hat{i}$ for the utterance $\hat{r}_{tr}$ as in Eq. (5.11)

**end for**

**Output:** The optimal system IDs for all evaluation utterances

is obtained from a universal background model; $\boldsymbol{T}$ is a low-rank rectangular matrix composed of basis vectors that span all variable spaces; and $\boldsymbol{z}^{(r)}$ is an i-vector for an utterance $r$.

Utterance similarities $\sigma$ are calculated from i-vectors for evaluation data $\boldsymbol{z}_{ev}^{(r_{ev})}$ and those for training data $\boldsymbol{z}_{tr}^{(r_{tr})}$. The most similar utterance $\hat{r}_{tr}$ to the evaluation data $r_{ev}$ is picked up from the training data as

$$\hat{r}_{tr} \leftarrow \arg\max_{r_{tr}} \sigma\left(\boldsymbol{z}_{ev}^{(r_{ev})}, \boldsymbol{z}_{tr}^{(r_{tr})}\right). \tag{5.9}$$

For similarity, e.g., cosine similarity (5.10) can be used.

$$\sigma\left(\boldsymbol{z}_{ev}^{(r_{ev})}, \boldsymbol{z}_{tr}^{(r_{tr})}\right) = \frac{\boldsymbol{z}_{ev}^{(r_{ev})} \cdot \boldsymbol{z}_{tr}^{(r_{tr})}}{\left|\boldsymbol{z}_{ev}^{(r_{ev})}\right| \left|\boldsymbol{z}_{tr}^{(r_{tr})}\right|}. \tag{5.10}$$

After the most similar utterance is found in the training data, the optimal system $\hat{i}$ is selected in the reference of WERs of training data as in Eq. (5.11) because similar utterances ought to have similar ASR performances.

$$\hat{i} \leftarrow \arg\min_{i} W_{tr}^{(\hat{r}_{tr})}(i). \tag{5.11}$$

Here, $W_{tr}^{(\hat{r}_{tr})}(i)$ is a WER of the $i$-th system for the utterance $\hat{r}_{tr}$[8]. Algorithm 4 shows the detailed procedure of the proposed method.

Fig. 5.4 shows an example of the proposed optimal system selection. In this case, there are two systems. First, respective WERs for all training data utterances are obtained. Next, for the given test data, i-vectors are calculated and the most similar utterance in the training data is found based on the i-vector similarity $\sigma$. In this case, the most similar utterance of the first utterance of the test data is the first utterance in the training data. Finally, in the reference of WER of the most similar utterance, the optimal system is selected. For the first utterance of the test data, the system two is selected because the WER of the second system is better than that of the first system.

[8] An average or an weighted average of WERs of the N-best results can be used for $W_{tr}$ instead of WERs of the 1-best results.

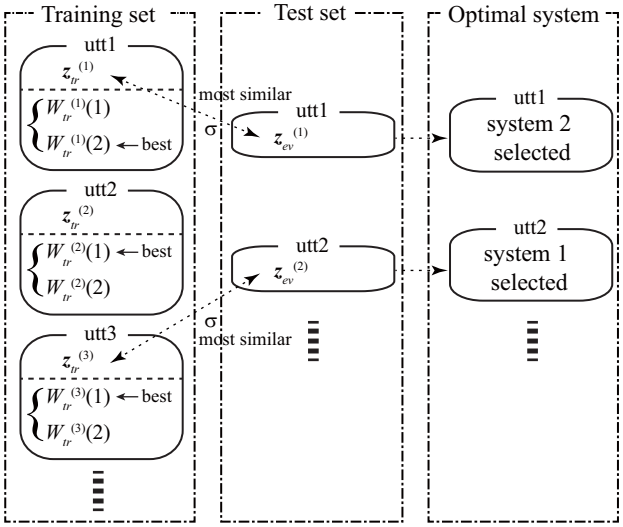



Fig. 5.4 Example of the proposed optimal system selection method.

### 5.3.4 Experimental setups

We validated the effectiveness of our proposed approach for the third CHiME challenge [278]. As mentioned in the introduction, this is a medium-vocabulary noisy ASR task whose speech utterances are taken from the *Wall Street Journal* database. There are two types of data: real data ("Real") and simulated data ("Sim."). The real data were recorded in the real world, whereas the simulated data were created by convolving clean speech with impulse responses and adding noise. Each type of data has four environments: bus, café, pedestrian, and street. WERs below are averaged over four environments. Table 5.10 shows the dataset description. The training set has 1,600 and 7,138 utterances by 4 and 83 speakers for Real and Sim, respectively. The development (Dev.) and evaluation (Eval.) set have 1,640 and 1,320 utterances, respectively, by 4 speakers both for Real and Sim. This section evaluated noisy speech, challenge-provided enhanced speech ("enh1"), and our enhanced speech ("enh2"). After multiple systems were constructed with two types of SE and various feature transformations, the optimal systems were selected by our proposed method or their hypotheses were combined by ROVER. Finally, language model scores were rescored by interpolating n-gram language model scores and recurrent neural network language model (RNN-LM) scores [209, 258]. The setups for RNN-LM were the same to those attached to the Kaldi WSJ example.

There were two types of acoustic feature settings. The first setting was MFCC with feature transformations. In addition to the standard 0–12th order MFCC features with $\Delta$ and $\Delta\Delta$, linear discriminant analysis (LDA) [125] compressed the static MFCCs in nine contiguous frames into 40-dimensional features before a global MLLT [126] was applied. The second setting started from the 0–22nd order fbank features with $\Delta$ and $\Delta\Delta$. For fMLLR, MLLT was used to de-correlate

Table 5.10 Number of utterances and speakers in each dataset of the third CHiME challenge.

| | # utterances | | # speakers | |
|---|---|---|---|---|
| dataset | Real | Sim. | Real | Sim. |
| Training set | 1,600 | 7,138 | 4 | 83 |
| Development set | 1,640 | 1,640 | 4 | 4 |
| Evaluation set | 1,320 | 1,320 | 4 | 4 |

Table 5.11 Setup for the ASR systems.

| | |
|---|---|
| Sampling frequency | 16 kHz |
| Window length | 25 ms |
| Window shift | 10 ms |
| Features (GMM) | 0–12th MFCCs + $\Delta$ + $\Delta\Delta$ |
| Features (DNN) | 0–22th filter banks + $\Delta$ + $\Delta\Delta$ |
| HMM states | 2,500 shared triphone states |
| Number of Gaussians | 15,000 |
| DNN nodes per layer | 1024 nodes |
| DNN layer size | 7 layers |
| Vocabulary size | 5,000 |

the features before adaptation. For both settings, to reduce the variances between speakers, SAT [135] was used where training is conducted after having transformed the training speech into a canonical space. The BN feature was a 40-dimensional hidden-layer unit output of DNN with two hidden layers. The warping parameters of linear VTLN were changed from 0.85 to 1.25 with a step of 0.01.

We trained DNNs after GMMs by using the Kaldi toolkit [124]. Table 5.11 shows the ASR setup. The detailed training procedure of GMMs was in [183, 218]. The number of monophones was 40, including silence. The number of context-dependent tri-phone states was 2,500 and the total number of Gaussians was 15,000. The parameters used in our experiments were the same to those in the challenge provided baseline. We used "nnet1" of the Kaldi toolkit for DNN training. Starting from the seven-layer restricted Boltzmann machine, the DNN was constructed where each hidden layer has a sigmoid activation. The learning rate was decreased from the initial learning rate (0.008) if the decrease of CE in the development set was under the threshold. Features across nine concatenated frames were inputted and the number of nodes per hidden layer was 1,024. We investigated the performance change when using feature-space boosted maximum mutual information (f-bMMI) [136] for GMM and sMBR for DNN.

### 5.3.5 Results and discussion

#### 5.3.5.1 GMM-based baseline ASR systems

Table 5.12 shows the average WER of GMM-based ASR systems on the Dev. and Eval. set. For all cases, SE improved the performance; "enh1" significantly improved the WER for Sim.

Table 5.12 Average WER [%] on the development and evaluation set of the third CHiME challenge using GMM acoustic models. The effectiveness of feature transformation and adaptation (FTA) and discriminative training (DT) is shown. Two types of SE methods (enh1 (challenge baseline) and enh2 (max SNR BF)) were evaluated in addition to noisy speech.

| | FTA | DT | Dev. set | | Eval. set | |
|---|---|---|---|---|---|---|
| | | | Real | Sim. | Real | Sim. |
| noisy | | | 26.90 | 24.40 | 43.06 | 30.70 |
| | ✓ | | 18.44 | 17.74 | 31.87 | 21.96 |
| | | ✓ | 16.04 | 14.78 | 27.05 | 17.16 |
| enh1 | | | 26.80 | 13.51 | 47.66 | 15.65 |
| | ✓ | | 19.92 | 9.76 | 35.78 | 11.16 |
| | | ✓ | 17.70 | **7.60** | 32.12 | **8.97** |
| enh2 | | | 21.35 | 16.51 | 36.49 | 22.77 |
| | ✓ | | 14.76 | 11.70 | 27.41 | 16.25 |
| | | ✓ | **12.43** | 9.05 | **21.61** | 13.33 |

but provided little improvement for Real. On the other hand, "enh2" significantly improved the WER for Real but was less effective for Sim. than "enh1". Feature transformation and adaptation (FTA in the figure) led to the WER improvement of 7–11%. From now on, the WER improvements were evaluated in terms of an absolute value. Discriminative training (DT in the figure) resulted in the additional WER improvements of approximately 2–3%. Even after SE, these techniques were still effective. These tendencies were similar to those of the second CHiME challenge [183, 218].

#### 5.3.5.2 DNN-based ASR systems

Table 5.13 shows the average WER of DNN-based ASR systems. The tendencies were similar to those in GMM-based systems. sMBR of DNN improved the WER by 1–2% especially effective for "enh2". fMLLR based model adaptation improved the WER by 1–3% but SAT was less effective (less than 1%). The BN feature was effective for Sim. but ineffective for Real. The VTLN provided an additional improvement on the Dev. set but worsened the WERs on Sim. of the Eval. set. These ASR systems were combined (Section 5.3.5.3) or selected (Section 5.3.5.4) because their performance tendencies were different from environment to environment.

#### 5.3.5.3 ASR system combination

Table 5.14 (C) shows the results of two, three, or six system combinations. Increasing the number of systems did not necessarily lead to the performance improvement because the best performing systems were different as shown in Table 5.13. For Dev. set, certainly, six system combination was the best for Sim. but for Real, three system combination was the best. For the reference, table also shows the WER of the best (B in the table) or the worst single system (W) from six systems. All systems were better than the worst system and some systems outperformed

Table 5.13 Average WER [%] on the development and evaluation set of the third CHiME challenge using DNN acoustic models.

| | BN | VTLN | fMLLR | SAT | sMBR | RNN-LM | Dev. set | | Eval. set | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Real | Sim. | Real | Sim. | |
| noisy | | | | | | | 15.58 | 13.51 | 29.21 | 18.41 | |
| | | | | | ✓ | | 14.41 | 12.62 | 28.49 | 16.90 | |
| | | | ✓ | | ✓ | | 12.11 | 11.40 | 22.74 | 13.57 | |
| | | | ✓ | ✓ | ✓ | | 12.05 | 11.16 | 22.25 | 13.95 | |
| | ✓ | | ✓ | ✓ | ✓ | | 12.31 | 10.75 | 22.91 | 12.67 | |
| enh1 | | | | | | | 17.64 | 7.44 | 32.03 | 9.04 | |
| | | | | | ✓ | | 16.51 | 7.01 | 30.84 | 8.26 | |
| | | | ✓ | ✓ | ✓ | | 13.65 | 6.04 | 24.32 | 7.04 | 1-a |
| | ✓ | | ✓ | ✓ | ✓ | | 14.42 | 5.92 | 26.17 | 6.33 | 1-b |
| | | ✓ | ✓ | ✓ | ✓ | | 13.11 | 5.90 | 20.22 | 12.45 | 1-c |
| | | | ✓ | ✓ | ✓ | ✓ | 11.88 | 4.65 | 21.66 | 5.22 | |
| | ✓ | | ✓ | ✓ | ✓ | ✓ | 12.78 | **4.41** | 24.11 | **4.75** | |
| | | ✓ | ✓ | ✓ | ✓ | ✓ | 11.36 | 4.57 | 17.93 | 10.23 | |
| enh2 | | | | | | | 12.83 | 9.38 | 25.94 | 14.57 | |
| | | | | | ✓ | | 11.36 | 8.39 | 22.41 | 12.84 | |
| | | | ✓ | ✓ | ✓ | | 9.03 | 7.08 | 16.98 | 10.45 | 2-a |
| | ✓ | | ✓ | ✓ | ✓ | | 9.67 | 6.89 | 17.74 | 9.99 | 2-b |
| | | ✓ | ✓ | ✓ | ✓ | | 13.97 | 6.11 | 20.02 | 13.69 | 2-c |
| | | | ✓ | ✓ | ✓ | ✓ | **7.39** | 5.69 | **14.79** | 8.65 | |
| | ✓ | | ✓ | ✓ | ✓ | ✓ | 8.02 | 5.48 | 15.59 | 8.19 | |
| | | ✓ | ✓ | ✓ | ✓ | ✓ | 12.18 | 4.76 | 17.64 | 12.08 | |

the best single system. This shows the effectiveness of system combination in exchange for the increase of computational resources. Rescoring with RNN-LM improved the WER further by 1–2%. Considering longer context than n-gram model was effective.

#### 5.3.5.4 Optimal ASR system selection

Our proposed method based on i-vectors selected the optimal single system from a combination of two types of SE methods and three types of feature transformations. Table 5.14 (S) shows the results. For Real, "enh1" tended to be picked up and for Sim. "enh2" tended to be picked up. All system selections were better than the worst system. This shows the effectiveness of the proposed method, because the proposed method aims to pick up the best system. The average differences between the best system –upper limit of a single system ASR– and the proposed system were 0.58% for Dev. set and 1.28% for Eval. set. In total, the worst WER of the selected system for either Real or Sim. was better than that of each single system. Tendencies were the same to the case of rescoring with RNN-LM. The average differences between the best system and the proposed system were 0.62% for Dev. set and 1.18% for Eval. set. The performance differences were larger for Eval. set than for Dev. set because Eval. set had larger mismatches between training and test data and the performance was worse.

Table 5.14 Average WER [%] on the development and evaluation set using system selection (S) and system combination (C). For reference, the best system (B) and the worst system (W) were picked up. Additionally, rescoring with RNN-LM was performed.

| Type | # of systems | Target systems | | | | | | RNN-LM | Dev. set | | Eval. set | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1-a | 1-b | 1-c | 2-a | 2-b | 2-c | | Real | Sim. | Real | Sim. |
| B | 1 from 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 9.03 | 5.90 | 16.98 | 6.33 |
| W | 1 from 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 14.42 | 7.08 | 26.17 | 13.69 |
| C | 2 | ✓ | | | ✓ | | | | 8.71 | 5.86 | 16.38 | 7.08 |
| C | 3 | ✓ | ✓ | ✓ | | | | | 12.73 | 5.51 | 19.59 | **6.28** |
| C | 3 | | | | ✓ | ✓ | ✓ | | **8.23** | 5.73 | 16.31 | 9.78 |
| C | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 10.02 | **5.27** | **15.67** | 7.42 |
| S | 1 from 2 | ✓ | | | ✓ | | | | *10.10* | 6.81 | 19.72 | 9.89 |
| S | 1 from 3 | ✓ | ✓ | ✓ | | | | | 14.24 | *5.98* | 25.67 | *7.35* |
| S | 1 from 3 | | | | ✓ | ✓ | ✓ | | 10.45 | 6.70 | *18.52* | 10.60 |
| S | 1 from 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 11.36 | 6.52 | 20.60 | 9.28 |
| B | 1 from 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 7.39 | 4.41 | 14.79 | 4.75 |
| W | 1 from 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 12.78 | 5.69 | 24.11 | 12.08 |
| C | 2 | ✓ | | | ✓ | | | ✓ | 7.52 | 4.59 | 14.61 | 5.72 |
| C | 3 | ✓ | ✓ | ✓ | | | | ✓ | 11.09 | 4.15 | 17.61 | **4.82** |
| C | 3 | | | | ✓ | ✓ | ✓ | ✓ | **6.66** | 4.54 | 14.15 | 8.39 |
| C | 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 8.70 | **3.93** | **13.74** | 5.97 |
| S | 1 from 2 | ✓ | | | ✓ | | | ✓ | *8.49* | 5.39 | 17.45 | 8.10 |
| S | 1 from 3 | ✓ | ✓ | ✓ | | | | ✓ | 12.49 | *4.55* | 23.06 | *5.49* |
| S | 1 from 3 | | | | ✓ | ✓ | ✓ | ✓ | 8.64 | 5.28 | *16.41* | 8.83 |
| S | 1 from 6 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 9.57 | 5.14 | 18.49 | 7.92 |

### 5.3.6 Conclusion

This section proposed an efficient optimal system selection method that estimates WERs of a test utterance based on the i-vector similarities when there are multiple ASR systems and their suitable environments are different. The proposed system selection can improve the worst performance for single systems by picking up better hypotheses. The experiments on the third CHiME challenge showed that the average differences between the best WER of the single system and that of the selected system were around 0.6% for the development set and 0.9% for the evaluation set. This shows the effectiveness of our proposed method. Our method does not increase the computational resources, although system combination improved the performance further but it increases the computational resources in proportion to the number of combined ASR systems. Future work will be a precise estimation of WER by using data clustering or an average of WERs of the N-best results.

## 5.4 Noisy ASR in public spaces 2 (The fourth CHiME challenge)

The fourth CHiME challenge is a revisit of the third CHiME challenge. The fourth one provides three tracks: 1ch, 2ch, and 6ch track [285]. 6ch track is the same setup of the third CHiME challenge. For all tracks, state-of-the-art baseline scripts were prepared. They employed discriminatively trained DNN acoustic models and RNN-based rescoring with advanced SE. There are four different environments in the tasks and for these kinds of tasks, system combination was effective. To realize more effective combination, we prepared multiple systems with different speech enhancement and different feature extractions. This section separately confirmed the effectiveness of our approach in terms of the WER.

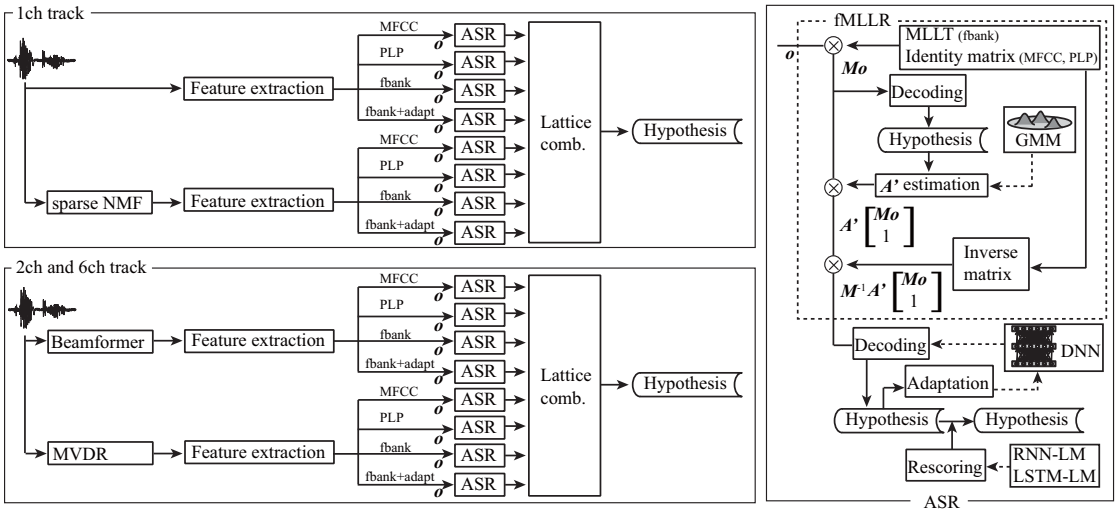



Fig. 5.5 Schematic diagram of the proposed ASR systems.

### 5.4.1 System overview

Fig. 5.5 shows the schematics of the proposed method. In each track, there were two types of speech enhancement. For each enhancement, three different features were used; and for FBANK feature, model-space speaker adaptation was performed. In total, hypotheses of eight systems are combined by using lattice combination.

#### 5.4.1.1 Front-end process

For single-channel track, sparse NMF [286] was used to suppress noise. To reduce distortions, enhanced speech was mixed with original noisy speech. For multi-channel track, in addition to the provided beamformer (BeamformIt), minimum variance distortionless response (MVDR) beamformer with precise steering vector estimation [92] was employed.

Table 5.15 System description for Table 5.16. All systems used DNN acoustic model.

| {m,p,f}-{s,m}-{n,s,b,m}-{u,a,a2}+{r,l} | |
|---|---|
| {m,p,f} | MFCC / PLP / fbank |
| {s,m} | Single / multi-channel data training |
| {n,s,b,m} | Noisy / sparse NMF / BeamformIt / MVDR |
| {u,a,$a_2$} | Unadapted / adapted / adapted-2 DNN |
| {r,l} | RNN / LSTM-LM rescoring |

#### 5.4.1.2 Back-end process

In addition to the provided 13-dimensional MFCC $+\Delta + \Delta\Delta$ with fMLLR transformation, we employed 13-dimensional PLP $+\Delta + \Delta\Delta$ with fMLLR transformation and 40-dimensional FBANK feature $+\Delta + \Delta\Delta$ with MLLT and fMLLR transformation [157]. Features in the consecutive 11 frames were input to the DNN.

In addition to the feature-space adaptation, model-space adaptation of DNN [287] was also used where the second layer of DNN was switched for each speaker. To train DNN acoustic models, multi-channel (6ch) data were all used whereas baseline only used single-channel data. These modification increased the training data size [92]. All training data were noisy without any speech enhancement, i.e., noisy data training. After decoding, we used long short-term memory (LSTM)-language model (LM) rescoring [288] instead of the baseline RNN-LM.

### 5.4.2 Experimental setups

The data were recorded by using hand-held tablets with six embedded microphones in four environments: bus (BUS), café (CAF), pedestrian (PED), and street (STR), with two types of data generation: data recorded in the real world (real) and data created by mixing real noise with clean speech recorded in a booth and convolved with measured impulse responses (simu). There are training, development (Dev), and test (Test) sets, and all the parameters for ASR were tuned on the Dev set.

We used the Kaldi toolkit [124]. The acoustic models were trained using the noisy data with no speech enhancement. The acoustic feature was the same as the challenge-provided one: the 13-dimensional MFCC $+\Delta + \Delta\Delta$ with fMLLR transformation. These features were obtained by a first-pass decoding using Gaussian mixture model systems, and the features in 11 consecutive frames were concatenated and used as an input to the DNN. After the second-pass decoding using DNN systems, we used a RNN-LM [209] for rescoring their hypotheses.

### 5.4.3 Results and discussion

Table 5.16 shows the WERs of the challenge. Descriptions of the system ID is shown in Table 5.15. Comparison of baseline1 and "m-m-n-u" shows the effectiveness of multi-channel data training, which was especially effective for 1ch track and improved the WERs by around

Table 5.16 Average WER [%] for the tested systems. For 1ch, "baseline1" was "m-s-n-u" and "baseline2" was "m-s-n-u+r". For 2ch and 6ch, "baseline1" was "m-s-b-u" and "baseline2" was "m-s-b-u+r". "best" combined asterisk-marked systems.

| Track | System | Dev | | Test | |
|---|---|---|---|---|---|
| | | real | simu | real | simu |
| 1ch | baseline1 | 14.67 | 15.67 | 27.69 | 24.15 |
| | baseline2 | 11.69 | 15.43 | 23.71 | 20.95 |
| | m-m-n-u | 12.67 | 13.55 | 22.17 | 20.29 |
| | m-m-n-u+l* | 7.76 | 8.92 | 15.66 | 15.12 |
| | p-m-n-u+l* | 7.74 | 9.23 | 16.03 | 15.31 |
| | f-m-n-u+l* | 5.60 | 7.60 | 11.76 | 12.75 |
| | f-m-n-a+l* | 5.58 | 7.70 | 11.85 | 12.72 |
| | m-m-s-u+l* | 7.78 | 8.86 | 15.49 | 15.08 |
| | p-m-s-u+l* | 7.60 | 9.33 | 15.47 | 15.61 |
| | f-m-s-u+l* | 5.56 | 7.30 | 11.64 | 12.76 |
| | f-m-s-a+l* | 5.41 | 7.48 | 11.64 | 12.90 |
| | best | 5.15 | 7.15 | 11.13 | 12.15 |
| 2ch | baseline1 | 10.90 | 12.36 | 20.44 | 19.03 |
| | baseline2 | 9.63 | 10.72 | 18.08 | 16.88 |
| | m-m-b-u | 9.90 | 10.60 | 16.89 | 16.27 |
| | m-m-b-u+l* | 5.59 | 6.33 | 11.43 | 10.55 |
| | p-m-b-u+l* | 5.51 | 6.48 | 11.71 | 10.77 |
| | f-m-b-u+l* | 4.19 | 5.23 | 8.38 | 9.10 |
| | f-m-b-a+l* | 3.96 | 5.15 | 8.23 | 8.49 |
| | m-m-m-u+l* | 5.34 | 6.09 | 11.21 | 11.55 |
| | p-m-m-u+l* | 5.03 | 6.40 | 11.11 | 11.61 |
| | f-m-m-u+l* | 3.96 | 5.23 | 8.45 | 9.62 |
| | f-m-m-a+l* | 3.80 | 5.06 | 7.99 | 9.10 |
| | best | 3.50 | 4.63 | 7.28 | 8.03 |
| 6ch | baseline1 | 8.14 | 9.07 | 15.04 | 14.20 |
| | baseline2 | 5.75 | 6.77 | 11.47 | 10.91 |
| | m-m-b-u | 7.69 | 8.23 | 12.57 | 12.66 |
| | m-m-b-u+r | 4.99 | 5.72 | 9.22 | 8.96 |
| | m-m-b-u+l* | 3.94 | 4.49 | 7.77 | 7.51 |
| | p-m-b-u+l* | 3.90 | 4.62 | 7.64 | 7.71 |
| | f-m-b-u+r | 4.18 | 4.95 | 7.20 | 7.47 |
| | f-m-b-u+l* | 3.10 | 3.63 | 5.94 | 6.28 |
| | f-m-b-a+l* | 3.05 | 3.60 | 5.71 | 5.94 |
| | m-m-m-u+r | 4.45 | 4.19 | 7.45 | 7.51 |
| | m-m-m-u+l* | 3.47 | 3.06 | 6.42 | 6.39 |
| | p-m-m-u+l* | 3.43 | 2.99 | 6.36 | 6.23 |
| | f-m-m-u+r | 3.72 | 3.66 | 6.11 | 6.67 |
| | f-m-m-u+l* | 2.75 | 2.61 | 5.19 | 5.72 |
| | f-m-m-a+l* | 2.60 | 2.53 | 5.06 | 5.01 |
| | f-m-m-$a_2$+l* | 2.47 | 2.45 | 4.75 | 4.39 |
| | best | 2.30 | 2.32 | 4.31 | 4.18 |

2–5%. Comparison of baseline1 and baseline2 and that of "m-m-n-u" and "m-m-n-u+l" show the effectiveness of LSTM-LM rescoring, which improved WER more than RNN-LM rescoring. The performances of MFCC and PLP features were almost equivalent but fbank feature significantly improved the WERs. DNN model adaptation was also effective. MVDR beamformer shows its

Table 5.17 WER [%] per environment for the best system.

| Track | Envir. | Dev | | Test | |
|---|---|---|---|---|---|
| | | real | simu | real | simu |
| 1ch | BUS | 7.15 | 6.24 | 18.00 | 8.55 |
| | CAF | 5.19 | 9.81 | 11.73 | 13.93 |
| | PED | 3.05 | 4.97 | 7.81 | 11.71 |
| | STR | 5.19 | 7.57 | 6.99 | 14.40 |
| 2ch | BUS | 4.54 | 3.92 | 11.42 | 5.08 |
| | CAF | 3.63 | 6.28 | 7.08 | 9.41 |
| | PED | 2.21 | 3.38 | 5.59 | 8.33 |
| | STR | 3.63 | 4.96 | 5.04 | 9.28 |
| 6ch | BUS | 3.07 | 2.01 | 5.16 | 2.95 |
| | CAF | 2.40 | 2.99 | 3.90 | 4.63 |
| | PED | 1.64 | 1.76 | 4.00 | 4.18 |
| | STR | 2.11 | 2.51 | 4.17 | 4.97 |

effectiveness for the 6ch track more than 2ch track, compared with the baseline beamformer. Combining multiple systems additionally improved WERs by around 0.3–0.6%. WERs of the best system were less than half of those of "baseline2" except one case (Test and simu in the 1ch track).

Table 5.17 shows the WER of the best system per environment in Table 5.16. Increasing the number of microphones was effective for all conditions. In real data, "BUS" was the most difficult task.

### 5.4.4 Conclusion

This section showed our approach for the fourth CHiME challenge. Multi-channel data training, fbank feature, and LSTM-LM based rescoring were the most effective. System combination gave additional improvements for all conditions.

## 5.5 Reverberant ASR in various rooms (The REVERB challenge)

The REverberant Voice Enhancement and Recognition Benchmark (REVERB) challenge is an Audio and Acoustic Signal Processing (AASP) challenge sponsored by the IEEE Signal Processing Society in 2013, and has recently been released for studying reverberant speech enhancement and recognition techniques [262]. This section focuses on the ASR task, which is a medium-sized vocabulary continuous ASR task, in order to evaluate its performance in reverberant environments.

In such a scenario, speech enhancement before ASR is important and impacts ASR performance. We have proposed a single-channel dereverberation method [127]. This method first estimates a reverberation time, which is one of the most important parameters for characterizing the extent of reverberation, and attempts to eliminate the reverberant components based on the estimated reverberation time. In addition, in order to exploit the eight-channel data provided by the REVERB challenge, we use a BF approach [265] with a direction-of-arrival estimation [29, 37].

In addition to the speech enhancement process, we focus on the state-of-the-art ASR techniques. The CHiME challenge and other existing evaluation campaigns for noise-robust ASR mainly focus on the variety of non-stationary additive noises, and the variety of room shapes or room types in these campaigns is very limited. On the other hand, the REVERB challenge [262] includes eight different reverberant environments: four rooms, which are composed of three simulated rooms and one real recorded room, multiplied by two types of source-to-microphone distances. In this scenario, due to the variety in the evaluation environments and the mismatch between simulated training data and real test data, discriminative training would cause over-training problems, although discriminative training is very powerful for matched conditions where training and evaluation conditions are close, in general. Therefore, it is important to confirm that ASR systems with discriminative training and feature transformations perform robustly in various reverberant environments.

This section deals with two feature transformation approaches: linear transformation and non-linear discriminative feature transformation. The former approach converts original feature vectors to new feature vectors based on linear transformation matrices (LDA and MLLT). LDA can reduce the influence of reverberation because the long context input features can handle the distorted speech features across several frames due to the influence of longer reverberation than the window size of the STFT [289, 290]. This property is particularly effective for reverberant ASR, and this section investigates the effectiveness of LDA on ASR performance in detail with varying context sizes. In addition, MLLT finds a linear transformation of features to reduce state-conditional feature correlations. For the latter approach, we use non-linear discriminative feature transformation [136], which directly reduces ASR errors by estimating non-linear feature transformation matrix with discriminative criteria.

The above feature transformation techniques estimate transformation matrices in the training

stage. However, to improve recognition accuracy for unknown conditions in the evaluation stage, the adaptation strategy of estimating feature transformation matrices for evaluation data is also effective. This section deals with basis fMLLR [291], which can estimate transformation matrices robustly even in the cases of short utterances. In addition, in the training stage, SAT [135] is also used. It trains acoustic models in a canonical speaker space based on the fMLLR framework in order to obtain better feature transformation in the adaptation stage.

After the feature transformations, GMM-based acoustic models are obtained by using discriminative training techniques [146, 147] and also this section deals with DNN [137]. Note that the lower layers of a DNN play the role of discriminative feature transformation [292], and our DNN system skips discriminative feature transformation, which is already included in a DNN.

The studies above mainly focus on a single ASR system. On the other hand, the use of multiple systems is another solution to improve the robustness of ASR performance [195, 197, 198]. For our proposed method, which exploits discriminative training methods, the best performing system is different from environment to environment due to the variety of evaluation data or mismatch between training and evaluation data. The system combination methods relax the degradation of speech recognition performance coming from these varieties or mismatches, e.g., [205, 293] proposed to use complementary system for system combination. This section constructs various systems that have different properties, and in particular, our proposed discriminative training method introduces complementary systems intentionally within a lattice-based discriminative training framework [256, 206]. The results from various recognizers will be combined using recognizer output voting error reduction (ROVER) [195].

In summary, there are three objectives in this section: First, the effectiveness of dereverberation and microphone-array speech enhancement techniques is validated. Second, the effectiveness of feature transformation and discriminative training for reverberant environments is validated. The objectives here are various types of acoustic modeling such as the GMM, subspace Gaussian mixture model (SGMM) [294], and DNN and their discriminative training. Third, to address the variety of reverberant environments, a system combination approach is introduced and its effectiveness is validated.

### 5.5.1 System overview

Fig. 5.6 shows a schematic diagram of the proposed system, which consists of three components. The first component is based on a speech enhancement step, which is described in Section 5.5.2. This section focuses on single- and eight-channel data. The speech enhancement part consists of (1) a multichannel delay-and-sum BF with direction-of-arrival estimation that enhances the direct sound compared with the reflected sound, (2) a single-channel dereverberation technique with reverberation time estimation that attempts to eliminate late reverberation, and (3) a normalized least-mean-squares (NLMS) adaptive filter algorithm that attempts to eliminate short-term distortions such as microphone difference or speech distortions caused by speech enhancement methods.

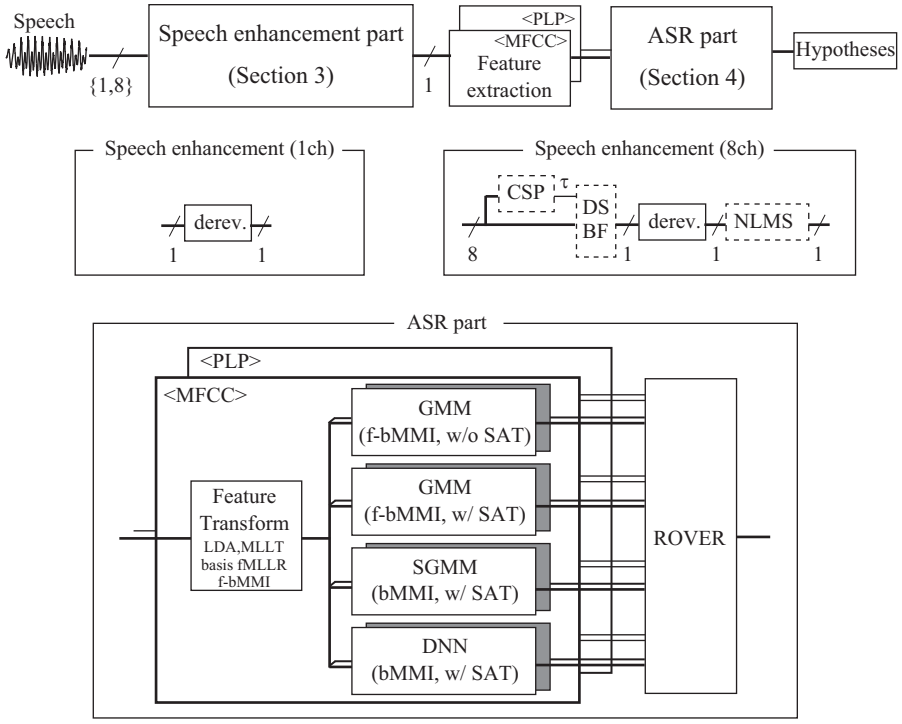



Fig. 5.6 Schematic diagram of the proposed system. (CSP: cross-spectrum phase analysis, DS-BF: delay-and-sum beamformer, derev.: proposed dereverberation method, and NLMS: normalized least-mean-squares adaptive filter algorithm.) Gray blocks are complementary systems for each system type.

The second component is based on a feature transformation step, including several feature-level transformations (LDA, MLLT, and basis fMLLR) and discriminative feature transformation (Section 4.4.6). This step uses two types of features: MFCC and PLP[9]. By using two different types of features, it is believed that complementary hypotheses can be obtained for system combination.

The third component is based on the ASR decoding step that uses a discriminatively trained acoustic model with margin control. Three types of systems (GMM, SGMM, and DNN) are constructed. Boosted maximum mutual information (bMMI) is used for GMM and SGMM in Sections 4.4.2 and for DNN in Section 4.4.4.

In addition to the three types of SAT model, a GMM model without SAT is also constructed; our proposed method constructed complementary systems for each system. The output results of 16 systems are combined using ROVER, and the final hypotheses are obtained.

[9]PLP features are effective for reverberant speech[295]

### 5.5.2 Speech enhancement

This section deals with speech enhancement methods: delay-and-sum BF with cross-spectrum phase (CSP) analysis in Section 5.5.3, a proposed dereverberation method in Section 3.2, and an NLMS algorithm that attempts to eliminate short-term distortion in Section 5.5.3.1. We describe them step by step. The delay-and-sum BF using the CSP method and NLMS adaptive filter algorithm is used for an 8-ch system; the dereverberation method is used for both the 1-ch and 8-ch systems.

### 5.5.3 Delay-and-sum BF with CSP analysis

To enhance the direct sound from the source, a frequency-domain delay-and-sum BF is applied [265]. The enhanced spectrum $\tilde{X}(t,k)$ is obtained by summing the spectrum $x_n(t,k)$ with a compensation of a time delay as

$$\tilde{X}(t,k) = \sum_n X_n(t,k) \cdot \exp\left(-2\pi\jmath \frac{k}{K}\tau_{t,n}\right). \tag{5.12}$$

The arrival time delay $\tau_{t,n}$ of the $n$-th microphone from the first microphone is related to the direction of arrival at the $t$-th frame (here $\tau_{t,1} = 0$). This time delay is estimated by CSP analysis as shown in Section 2.2.3.

#### 5.5.3.1 NLMS adaptive filter algorithm

The goal of the NLMS adaptive filter algorithm is to eliminate short-term distortions from an observed distorted signal sequence $\boldsymbol{z}_s = [z(s-N_L+1),\ldots,z(s)]^\top \in \mathbb{R}^{N_L}$ based on a desired signal $d_s$ [296] by using a linear filter with the tap length $N_L$. Filters $\boldsymbol{w}'_s \in \mathbb{R}^{N_L}$ that realize these requirements are recursively trained in a manner where errors between filtered signals and desired signals are minimized as

$$\min_{\boldsymbol{w}'_s} |d_s - \boldsymbol{z}_s^\top \boldsymbol{w}'_s|^2. \tag{5.13}$$

An LMS algorithm uses instantaneous values for the estimation of a gradient, and an NLMS algorithm normalizes the step size parameter by the signal power. Thus, the update formula of an NLMS algorithm is obtained as

$$\boldsymbol{w}'_s = \boldsymbol{w}'_{s-1} + \frac{\varrho}{\epsilon + |\boldsymbol{z}_s|^2}\boldsymbol{z}_s[d_s - \boldsymbol{z}_s^\top \boldsymbol{w}'_{s-1}], \tag{5.14}$$

where $\varrho$ is a step size, and $\epsilon$ is a very small constant that avoids the instability of the update formula. The initial value of filter $\boldsymbol{w}'_0$ is 0. In this case, $\boldsymbol{z}_s$ is a reverberant speech, and $d_s$ is a clean speech without reverberation. A filter $\boldsymbol{w}'$ is obtained from the entire training data set. For evaluation, desired signals $d_s$ cannot be obtained; thus, the filter cannot be changed. The tap length of NLMS is short because the goal of this filter is to eliminate a short-term distortion, whereas the proposed dereverberation algorithm (3.2) attempts to eliminate late reverberation.

## 5.5.4 Speech recognition

### 5.5.4.1 Basis fMLLR

For adaptation, instead of a normal fMLLR transformation, the basis fMLLR [291] is used. It can robustly estimate transform matrices and bias terms even for short utterances. This method realizes the transformation of original features $\boldsymbol{y}'_t$ into adapted features $\boldsymbol{y}''_t$ by using pretrained bases of transform matrices and bias terms and estimating their weights as

$$\boldsymbol{y}''_t = \sum_{\nu} \pi_\nu \left[ \boldsymbol{A}^f_\nu \boldsymbol{y}'_t + \boldsymbol{b}^f_\nu \right], \tag{5.15}$$

where $\boldsymbol{A}^f_\nu \in \mathbb{R}^{I' \times I'}$ and $\boldsymbol{b}^f_\nu \in \mathbb{R}^{I'}$ are the $\nu$-th pre-trained basis of an fMLLR transform matrix and bias term, respectively, which are estimated from entire training data. For evaluation, only their weights $\pi_\nu$ are estimated.

Moreover, to address the wide variety between speakers, SAT as an acoustic model adaptation [135] is frequently used. In SAT training, acoustic models are trained on speaker-adapted training data, which are transformed into canonical speaker space by using speaker adaptation techniques, in this case, fMLLR. This can reduce the influence of a speaker variation. This section validates the effectiveness of feature transformations (LDA and MLLT) and adaptation techniques (basis fMLLR and SAT).

### 5.5.4.2 Discriminative training of acoustic models and feature transformation

This section compares the performances of bMMI training of GMM and SGMM (Section 4.4.3) to those of maximum ML training and bMMI of DNN (Section 4.4.4) to those of CE training. In addition, we validate the effectiveness of a f-bMMI (Section 4.4.6) and our proposed discriminative trained complementary system (Section 4.6.2) suitable for system combination.

## 5.5.5 Experimental setups

We validated the effectiveness of our proposed approaches for a reverberated speech recognition task on the REVERB challenge data. The task is a medium-vocabulary ASR in reverberant environments, whose utterances are taken from the *Wall Street Journal* (*WSJ*) database (WSJ-CAM0 [297]). This database includes two types of data: SimData created by convolving clean speech with six types of room impulse responses at a distance of 0.5m (near) or 2m (far) from the microphones in three offices (Rooms 1, 2, and 3) whose reverberation times are 0.25, 0.5, and 0.75s, respectively, with relatively stationary noise at 20dB SNR; and RealData created by recording real-world speech at a distance of 1m or less (near) or 2.5m or less (far) from the microphones in one room (Room 1) with stationary noise such as air conditioner noise. Eight microphones were arranged on the circle with a radius of 0.1m. The number of speakers and

Table 5.18 Number of speakers and utterances of training (**tr**), development (**dev**), and evaluation (**eva**) set for the REVERB challenge.

| set | | Number of speakers | Number of utterances |
|---|---|---|---|
| **tr** | - | 92 | 7,861 |
| **dev** | SimData | 20 | 1,484 |
| | RealData | 5 | 179 |
| **eva** | SimData | 28 | 2,176 |
| | RealData | 10 | 372 |

Table 5.19 Category of the REVERB challenge speech recognition task. Underline "_" denotes the category to which this section belongs.

| | Type |
|---|---|
| Processing Scheme | full batch, utterance-based, real-time |
| Training data of acoustic model | own dataset, multi-condition, clean |
| Recognizer type | own recognizer, challenge baseline recognizer |
| Number of channels used | 1, 2, 8 |

utterances of the training set (**tr**), evaluation set (**eva**), and development set (**dev**) is shown in Table 5.18.

Acoustic models were trained using **tr**. Some of the parameters, e.g., language model weights, were tuned based on the WERs of **dev**. The vocabulary size is 5 k, and a trigram language model is used. The REVERB challenge speech recognition task is categorized in terms of processing techniques, training data of the acoustic model, recognizer type, and number of channels used, as shown in Table 5.19. All experiments in this section were "utterance-based batch processing,"[10] "acoustic model trained on the challenge provided multicondition (MC) training data," "own recognizer," and "single- or eight-channel data." These systems were constructed by using the Kaldi toolkit [124].

#### 5.5.5.1 Speech enhancement

The REVERB challenge provides single-, two-, and eight-channel data. We used single- and eight-channel data. For single- and eight-channel data, the proposed dereverberation technique was used with parameters: $D = 9$, $\alpha = 5$, $\beta = 0.05$, $a = 0.005$, and $b = 0.6$. For eight-channel data, before dereverberation, delay-and-sum BF with a direction of arrival estimation by CSP analysis was performed, which used a total of ${}_8C_2 (= 28)$ pairs of microphones. To improve the performance of the original CSP method, we used a peak-hold process [44] and noise component suppression, which sets the cross power spectrum to zero when the estimated signal-to-noise

[10] This allows for multiple decoding passes per utterance, such as for calculating the fMLLR matrix, but decodes each test utterance separately, without taking into account information from other test utterances, or speaker identities.

ratio (SNR) is below 0dB [37]. Synchronous addition of multiple microphone pair-wise CSP coefficients reduces the noise influence [35]. After dereverberation, NLMS adaptive filters with $N_L = 200$ taps were applied.

#### 5.5.5.2 Feature extraction and transformation and acoustic model adaptation

We describe the settings of acoustic features and feature transformations, which are detailed in [218, 183]. The baseline acoustic features were 0–12 order MFCCs and PLPs with first and second dynamic features. After concatenating static MFCCs/PLPs during $L + R + 1$ frames without using delta feature, a total of $(13 \times (L + R + 1))$-dimensional features were compressed into 40 dimensions by the LDA.

For adaptation, when speaker IDs were known for the training set, bases $\boldsymbol{A}_\nu^f$ and $\boldsymbol{b}_\nu^f$ were estimated. For the development and evaluation set, speaker IDs are assumed to be unknown, and weight vector $\pi_\nu$ was estimated.

#### 5.5.5.3 Discriminative methods

In discriminative feature transformation (Section 4.4.6), a UBM with $N_g = 400$-mix Gaussians was used. The offset features were calculated for each composed of 40-dimensional features, including MFCC/PLP features with dynamic features (39 dimensions in total) and the posterior probability of it, with context expansion (contiguous nine frames). The number of dimensions of feature vector $\boldsymbol{h}_t$ was 400[Gauss] $\times$ 40[dim/(Gauss $\cdot$ frame)] $\times$ 9[frame]. Features with the top two GMM posteriors were selected and all other features were ignored.

The boosting factor $b$ of bMMI and f-bMMI was 0.1. To construct complementary systems, the additional boosting factor $b_1$ in the second term of Eq. (4.57) was 0.3 and $\alpha_c$ was 0.75. For f-bMMI, in one iteration, f-bMMI for the matrix $\boldsymbol{M}$ was coupled with bMMI for the acoustic model parameters $\lambda$.

#### 5.5.5.4 Building acoustic models

First, clean acoustic models were trained. The number of monophones was 45, including silence ("sil"). Triphone model has 2,500 states and 15,000 Gaussian distributions. Second, using the alignments and triphone tree structures of the clean model, reverberated acoustic models were trained on the MC dataset according to the ML criterion. Finally, from this ML model, we performed the discriminative training and feature transformations.

For DNNs, we used Povey's implementation of neural network training in Kaldi [124]. DNN has two hidden layers was two and each hidden layer has 642 nodes. The total number of parameters was 2M. The initial learning rate of CE training was 0.02, and this decreased to 0.004 at the end of training. The training targets for the DNN were determined by the forced alignments on reverberant speech using a GMM model with SAT. The parameters used in our experiments

were set as those in the *WSJ* tutorial (s6) attached to the Kaldi toolkit, although some settings such as the number of model parameters or some minor parameters were modified.

#### 5.5.5.5 System combination

We prepared three types of ASR acoustic model systems for the challenge: GMM, SGMM, and DNN. To improve the performance of the respective systems, for GMM, f-bMMI was used; whereas for SGMM and DNN, bMMI was used. On the development set, because output tendencies of GMM with and without SAT model were different, both systems were used for a system combination. For each system, complementary systems were constructed by the proposed method as shown in 4.6.1. These systems were trained both for MFCC and PLP features; thus, a total of 16 systems were prepared. After decoding for generated lattices, minimum Bayes risk decoding [268], which slightly improved the performance, was commonly used.

#### 5.5.5.6 Black-box optimization

Bayesian optimization using Gaussian processes [298] was applied to various speech recognition problems including neural network [299] and HMM topology optimization [300]. In this section, we also applied this technique to the selection of combined systems and the parameter optimization for ROVER. The objective function of the optimization was WER of the development set.

### 5.5.6 Results and discussion

#### 5.5.6.1 Baseline and speech enhancement techniques

Tables 5.20 and 5.21 show the WERs of the development set (**dev**) for three simulated rooms and one real room with two types of source-to-microphone distances (near/far). Table 5.20 is based on a single-channel one and Table 5.21 is based on an eight-channel one. The "Kaldi baseline" in Table 5.20 is an acoustic model trained on the MC data without speech enhancement. "derev." is the proposed dereverberation method with a reverberation time estimation. Although, for some cases in Room 1, the reverberation time is fairly short and the proposed method degraded performance, for other cases and on average, performance was improved by approximately 2%. [301] showed that our proposed dereverberation technique is effective even with a state-of-the-art de-noising auto-encoder. For the eight-channel data shown in Table 5.21, BF with "derev." significantly improved performance by approximately 6.3–8.3% on average, because the direction of arrival estimation was stable and reliable. "NLMS" improved the WER by 2.0% for the RealData, but degraded the WER by 0.6% for the SimData. However, because these decreases in performance has less impact than the improvements, we used NLMS below.

Table 5.20 WER [%] in terms of rooms and microphone distances on the REVERB challenge **dev** set using single-channel data and MFCC features. The proposed dereverberation method was used. Three types of acoustic models (GMM, SGMM, and DNN) were constructed with feature transformation (LDA+MLLT), adaptation (basis fMLLR and SAT), and discriminative training (bMMI and f-bMMI).

| | | | SimData | | | | | | | RealData | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Room 1 | | Room 2 | | Room 3 | | Avg | Room 1 | | Avg |
| | Feature | Type | near | far | near | far | near | far | | near | far | |
| Kaldi baseline | MFCC | ML | 10.96 | 12.56 | 15.70 | 34.21 | 19.61 | 39.24 | 22.05 | 48.53 | 47.37 | 47.95 |
| derev. | | | 12.41 | 14.68 | 14.03 | 27.16 | 16.39 | 33.85 | 19.75 | 47.04 | 44.57 | 45.81 |
| GMM | +LDA+MLLT | ML | 9.46 | 11.01 | 11.51 | 22.04 | 13.08 | 28.09 | 15.87 | 39.99 | 40.67 | 40.33 |
| | +basis fMLLR | | 7.77 | 10.00 | 9.76 | 19.28 | 11.05 | 24.90 | 13.79 | 33.00 | 35.54 | 34.27 |
| | | bMMI | 7.13 | 9.61 | 9.12 | 16.19 | 10.46 | 21.98 | 12.42 | 30.69 | 35.20 | 32.95 |
| | | f-bMMI | 6.27 | 8.73 | 8.28 | 14.89 | 9.37 | 19.54 | 11.18 | **28.32** | **31.31** | **29.82** |
| | | f-bMMI$_c$ | 7.06 | 9.05 | 8.58 | 14.96 | 10.16 | 20.43 | 11.71 | 29.01 | 31.72 | 30.37 |
| | +SAT | ML | 8.87 | 11.21 | 9.71 | 19.89 | 10.95 | 24.04 | 14.11 | 36.06 | 36.23 | 36.15 |
| | | bMMI | 6.56 | 8.51 | 7.76 | 16.24 | 9.03 | 19.88 | 11.33 | 34.19 | 37.53 | 35.86 |
| | | f-bMMI | 5.88 | 7.60 | 7.25 | 14.59 | 8.09 | 17.51 | 10.15 | 31.63 | 34.72 | 33.18 |
| | | f-bMMI$_c$ | 6.07 | 7.82 | 7.22 | 14.89 | 8.43 | 17.51 | 10.32 | 32.38 | 35.27 | 33.83 |
| SGMM | | ML | 6.47 | 9.07 | 8.18 | 17.11 | 9.55 | 20.40 | 11.80 | 33.13 | 34.93 | 34.03 |
| | | bMMI | 5.53 | 7.23 | 7.00 | 14.44 | 7.76 | 17.48 | 9.91 | 31.50 | 33.36 | 32.43 |
| | | bMMI$_c$ | 5.68 | 7.28 | 7.02 | 14.44 | 7.94 | 17.68 | 10.01 | 30.94 | 33.08 | 32.01 |
| DNN | | CE | 6.71 | 8.85 | 8.70 | 15.58 | 9.15 | 19.07 | 11.34 | 30.88 | 35.82 | 33.35 |
| | | bMMI | 5.29 | 7.06 | 6.95 | 13.09 | 7.57 | 15.53 | 9.25 | 28.45 | 32.67 | 30.56 |
| | | bMMI$_c$ | **5.14** | **6.74** | **6.51** | **12.37** | **7.27** | **15.50** | **8.92** | **28.32** | 33.49 | 30.91 |

These results above used MFCC features. Experimental results using PLP features are shown in Table 5.22. On average, the ASR performances using PLP features were approximately 0.2–1% lower than those using MFCC features; however, their error tendencies were fairly different, which was a good property for system combination.

#### 5.5.6.2 LDA and MLLT feature transformation and adaptation

LDA and MLLT feature transformations significantly improved performance by approximately 2.6–5.5%. Table 5.23 shows the effect of an LDA context size on performance. The performance of the SimData could not be improved by context sizes longer than 4. For the RealData, performance could be improved in several cases by adding more right context, but generally not by adding left context. In reverberant environments, because reverberant components of current frames give an influence on the features in the right context, the right context can be useful for improving speech recognition performance. In the end, we kept the context size at the default setting, $L = R = 4$.

Tables 5.20 and 5.21 show that the adaptation technique, basis fMLLR, improved performance by approximately 1.3–6.9%. The effect of SAT is unstable between environments.

Table 5.21 WER [%] on the REVERB challenge **dev** set using eight-channel data and MFCC features. In addition to the proposed dereverberation method, BF with direction of arrival estimation by CSP analysis and NLMS adaptive filters were used.

| | | | SimData | | | | | | | RealData | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Room 1 | | Room 2 | | Room 3 | | Avg | Room 1 | | Avg |
| | Feature | Type | near | far | near | far | near | far | | near | far | |
| CSP+BF+derev. | MFCC | ML | 10.79 | 12.19 | 11.02 | 16.71 | 11.47 | 20.43 | 13.77 | 40.36 | 42.83 | 41.60 |
| +NLMS | | | 11.11 | 12.27 | 11.81 | 17.40 | 12.34 | 21.46 | 14.40 | 38.37 | 40.74 | 39.56 |
| GMM | +LDA+MLLT | ML | 8.38 | 10.30 | 9.91 | 14.94 | 10.19 | 17.28 | 11.83 | 34.06 | 37.18 | 35.62 |
| | +basis fMLLR | | 7.74 | 9.22 | 8.80 | 13.33 | 9.05 | 15.28 | 10.57 | 27.39 | 30.14 | 28.77 |
| | | bMMI | 6.64 | 8.21 | 7.25 | 11.39 | 7.10 | 11.50 | 8.68 | 24.89 | 27.96 | 26.43 |
| | | f-bMMI | 6.19 | 7.40 | 7.39 | 10.13 | 6.58 | 10.24 | 7.99 | **22.58** | **26.25** | **24.42** |
| | | f-bMMI$_c$ | 6.39 | 7.33 | 7.44 | 9.86 | 6.70 | 10.44 | 8.03 | 22.71 | 27.41 | 25.06 |
| | +SAT | ML | 7.25 | 9.32 | 8.70 | 12.79 | 8.33 | 13.80 | 10.03 | 28.88 | 32.88 | 30.88 |
| | | bMMI | 5.24 | 7.10 | 6.56 | 9.93 | 5.98 | 10.98 | 7.63 | 26.58 | 30.83 | 28.71 |
| | | f-bMMI | 5.01 | 6.76 | **5.96** | **9.07** | 5.84 | **9.40** | 7.01 | 24.27 | 29.60 | 26.94 |
| | | f-bMMI$_c$ | 5.16 | 6.93 | 6.11 | 9.49 | 5.96 | 9.67 | 7.22 | 24.27 | 29.73 | 27.00 |
| SGMM | | ML | 5.65 | 7.62 | 7.47 | 10.97 | 7.00 | 11.45 | 8.36 | 25.27 | 30.35 | 27.81 |
| | | bMMI | **4.57** | **6.05** | 6.19 | 9.27 | 6.01 | 9.89 | **7.00** | 24.70 | 30.01 | 27.36 |
| | | bMMI$_c$ | 4.72 | 6.10 | 6.09 | 9.56 | 6.18 | 10.01 | 7.11 | 24.39 | 30.01 | 27.20 |
| DNN | | CE | 6.49 | 7.45 | 7.84 | 11.44 | 7.25 | 11.97 | 8.74 | 25.27 | 29.32 | 27.30 |
| | | bMMI | 5.56 | 6.27 | 6.24 | 9.29 | 5.71 | 10.44 | 7.25 | 23.27 | 28.84 | 26.06 |
| | | bMMI$_c$ | 5.26 | **6.05** | 6.21 | 9.10 | **5.61** | 10.06 | 7.05 | 22.65 | 28.50 | 25.58 |

#### 5.5.6.3 Discriminative training of acoustic model and discriminative feature transformation

Tables 5.20 and 5.21 show that the discriminative training was effective for reverberant environments. The performances of f-bMMI training were higher than those of bMMI training in all cases by approximately 0.6–1.7%. The WERs of our complementary systems were only slightly lower (0.2–0.7%) than those of the base systems, and they have different tendencies from base systems; thus, they appear to be well suited to system combination.

Table 5.24 shows the effect of the iteration numbers of bMMI and f-bMMI on the development set performance. The results show that the best performance was achieved at four iterations.

#### 5.5.6.4 SGMM and DNN

Tables 5.20 and 5.21 show the performance of SGMM acoustic models. For the SimData, the performance of SGMMs was higher than that of GMMs. However, for the RealData, the performance was lower than that of GMMs. Because the RealData were noisier than the SimData, the estimation of speaker vector can be unstable.

DNN acoustic models achieved the best performance for the SimData. Although the best system for the RealData was GMM without SAT, DNN was the second best. On average over the SimData and RealData, DNNs achieved the best performance. Although DNN was trained discriminatively even by CE training according to the frame-level discriminative criterion,

Table 5.22 Average WER [%] on the REVERB challenge **dev** set using PLP features.

| | | | 1ch | | 8ch | |
|---|---|---|---|---|---|---|
| | Feature | | SimData | RealData | SimData | RealData |
| Kaldi baseline | PLP | ML | 22.96 | 48.90 | | |
| derev. | | | 19.84 | 44.15 | | |
| CSP+BF+derev. | | | | | 13.98 | 42.21 |
| +NLMS | | | | | 14.97 | 41.15 |
| GMM | +LDA+MLLT | ML | 15.63 | 40.36 | 12.13 | 35.11 |
| | +basis fMLLR | | 13.70 | 34.21 | 10.73 | 29.21 |
| | | bMMI | 12.78 | 33.43 | 8.94 | 26.84 |
| | | f-bMMI | 11.91 | 30.67 | 8.10 | **25.72** |
| | | f-bMMI$_c$ | 12.20 | 31.67 | 8.26 | 26.30 |
| | +SAT | ML | 13.55 | 36.25 | 10.17 | 30.85 |
| | | bMMI | 11.05 | 35.63 | 8.06 | 28.45 |
| | | f-bMMI | 10.14 | 33.29 | 7.32 | 26.78 |
| | | f-bMMI$_c$ | 12.20 | 31.67 | 7.61 | 27.59 |
| SGMM | | ML | 11.90 | 32.95 | 8.43 | 26.99 |
| | | bMMI | 10.25 | 33.10 | 7.13 | 26.67 |
| | | bMMI$_c$ | 10.30 | 33.14 | 7.19 | 27.21 |
| DNN | | CE | 11.30 | 31.87 | 8.75 | 27.33 |
| | | bMMI | 9.44 | 30.19 | 7.25 | 26.06 |
| | | bMMI$_c$ | **9.40** | **30.13** | **6.74** | 26.37 |

Table 5.23 Average WER[%] investigating the effect of LDA context sizes [left ($L$) and right ($R$)] on the REVERB challenge **dev** set using eight-channel data.

| | SimData | | | | RealData | | | |
|---|---|---|---|---|---|---|---|---|
| $L \setminus R$ | 4 | 5 | 6 | 7 | 4 | 5 | 6 | 7 |
| 4 | **11.83** | 12.20 | 12.10 | 12.57 | 35.62 | 34.31 | 34.10 | 36.22 |
| 5 | 12.14 | 12.32 | 12.46 | 12.72 | 34.71 | 35.34 | 34.44 | **33.31** |
| 6 | 12.57 | 12.33 | 12.56 | 12.87 | 35.49 | 35.29 | 34.19 | 35.11 |
| 7 | 12.83 | 12.94 | 13.43 | 13.49 | 35.13 | 35.90 | 35.67 | 36.00 |

Table 5.24 Average WER [%] investigating the effect of iteration numbers of bMMI and f-bMMI discriminative training with SAT on the REVERB challenge **dev** set using eight-channel data.

**bMMI**

| | MFCC | | | | PLP | | | |
|---|---|---|---|---|---|---|---|---|
| # of iterations | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| SimData | 8.70 | 8.41 | 8.18 | **7.63** | 9.02 | 8.64 | 8.47 | **8.06** |
| RealData | 29.21 | 28.34 | **28.16** | 28.71 | 29.74 | 29.26 | 28.91 | **28.45** |

**f-bMMI**

| | MFCC | | | | PLP | | | |
|---|---|---|---|---|---|---|---|---|
| # of iterations | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| SimData | 8.07 | 7.56 | 7.30 | **7.01** | 8.47 | 7.93 | 7.57 | **7.32** |
| RealData | 27.70 | 27.29 | 27.16 | **26.94** | 29.36 | 27.86 | 27.15 | **26.78** |

sequence discriminative training, bMMI, for DNN systems turned out to be as effective as for other systems.

Table 5.25 WER [%] on the REVERB challenge **dev** set, with system combination using both MFCC and PLP features. For GMM systems, f-bMMI is used, while for SGMM and DNN systems, bMMI is used. The number 2 stands for MFCC and PLP systems, and the number 4 stands for MFCC and PLP systems along with their complementary systems. ROVER 6) uses black-box optimization at the stage of system selection and parameter optimization for ROVER.

| | | | | | | SimData | | | | | | | RealData | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Number of systems | | | | Room 1 | | Room 2 | | Room 3 | | Avg | Room 1 | | Avg |
| | ID | GMM | SAT-GMM | SGMM | DNN | near | far | near | far | near | far | | near | far | |
| 1ch | 1) | | 2 | | | 6.00 | 8.19 | 7.52 | 14.37 | 8.78 | 18.35 | 10.54 | 27.70 | 30.35 | 29.03 |
| | 2) | 2 | 2 | | | 5.31 | 6.37 | 6.58 | 12.62 | 7.42 | 16.00 | 9.05 | 27.26 | 29.60 | 28.43 |
| | 3) | 4 | 4 | | | 5.33 | 6.39 | 6.63 | 12.67 | 7.49 | 15.60 | 9.02 | 27.01 | 29.67 | 28.34 |
| | 4) | 4 | 4 | 4 | | 5.01 | 6.34 | 6.33 | 12.45 | 6.87 | 15.43 | 8.74 | 26.64 | 29.80 | 28.22 |
| | **5)** | 4 | 4 | 4 | 4 | 4.67 | 5.88 | 6.31 | **11.93** | 6.63 | **14.89** | 8.39 | 26.58 | **28.91** | **27.75** |
| | 6) | 2 | 2 | 2 | 2 | **4.52** | **5.68** | **6.29** | 12.00 | **6.50** | 15.06 | **8.34** | **26.45** | 29.80 | 28.13 |
| 8ch | 1) | | 2 | | | 4.72 | 5.83 | 5.96 | 8.92 | 5.37 | 8.75 | 6.59 | 23.27 | 28.30 | 25.79 |
| | 2) | 2 | 2 | | | 4.72 | 6.02 | 5.72 | 8.26 | 5.14 | 8.56 | 6.40 | 22.27 | 26.59 | 24.43 |
| | 3) | 4 | 4 | | | 4.72 | 5.83 | 5.77 | 8.21 | 5.19 | 8.38 | 6.35 | 22.52 | 26.52 | 24.52 |
| | 4) | 4 | 4 | 4 | | **4.08** | 5.16 | 5.62 | 7.79 | **4.80** | 8.38 | 5.97 | 22.40 | 27.00 | 24.70 |
| | **5)** | 4 | 4 | 4 | 4 | 4.18 | **5.11** | **5.50** | **7.74** | 4.85 | **8.23** | **5.94** | 21.90 | 26.52 | 24.21 |
| | *6)* | 3 | 1 | 4 | 2 | 4.18 | 5.51 | **5.50** | **7.74** | 4.97 | 8.43 | 6.06 | **21.58** | **26.32** | **23.95** |

#### 5.5.6.5 System combination

We tested five types of system combinations, as shown in Table 5.25. The number 2 stands for one MFCC system and one PLP system. The number 4 stands for two MFCC and two PLP systems composed of a base system and the proposed complementary system. These systems' outputs are combined by using ROVER. The ID 1) system was a combination of SAT-GMMs (f-bMMI) using both MFCC and PLP features. The performance for the RealData improved by 1.2–4.2% over the f-bMMI with a SAT (MFCC) single system. For the GMM system without SAT, using f-bMMI [ID 2)], the WER improved by 0.2–1.5% for the SimData and 0.6–1.4% for the RealData, respectively. Including the complementary systems [ID 3)], the WER improved slightly. For the best case, WER improved by 0.4%, while for the worst case, WER decreased by 0.1%. This shows the effectiveness of our proposed method. Adding in SGMMs [ID 4)], which was effective for the SimData, the performance for the SimData further improved by 0.3–0.4%. Taking into account DNNs [ID 5)], the performance was again improved; this system, which combined 16 systems in total, achieved the best average performance on the development set. For the reference, the results of eight systems combination without using our proposed combination are added to the last line of 1ch case [ID 6)]. The WER on RealData was worse than those of the proposed 16 system combination, which shows that the complementary training generalizes the ASR results for unseen data conditions more.

In all cases except for the Room 1/far(8-ch) condition,[11] the performances were better than those of the best system. This shows that the system combination approach is effective for the case where reverberant environments are various.

[11] In this case, GMM(f-bMMI) exhibited the best performance (26.25% WER).

Table 5.26 WER [%] on the REVERB challenge **eva** set. All systems except ROVER are single systems. MFCC feature was used for single system, and MFCC and PLP features were used for ROVER 5).

| | | SimData | | | | | | | RealData | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Room 1 | | Room 2 | | Room 3 | | Avg | Room 1 | | Avg |
| | | near | far | near | far | near | far | | near | far | |
| 1ch | Kaldi baseline | 13.23 | 14.13 | 15.54 | 29.69 | 20.06 | 37.44 | 21.68 | 50.62 | 45.98 | 48.30 |
| | derev. | 12.50 | 13.43 | 14.61 | 24.71 | 17.09 | 32.62 | 19.16 | 44.75 | 43.32 | 44.04 |
| | GMM (f-bMMI) | 7.27 | 8.17 | 8.82 | 14.11 | 10.54 | 18.76 | 11.28 | 28.65 | 29.54 | 29.10 |
| | GMM (SAT,f-bMMI) | 6.44 | 7.22 | 7.57 | 13.97 | 9.52 | 18.44 | 10.53 | 28.87 | 29.78 | 29.33 |
| | SGMM (SAT, bMMI) | 5.81 | 6.54 | 7.22 | 13.84 | 8.70 | 18.17 | 10.05 | 27.75 | 28.36 | 28.06 |
| | DNN (SAT, bMMI) | 5.90 | 6.84 | 7.35 | 12.57 | 9.40 | 16.55 | **9.77** | 25.97 | 25.69 | **25.83** |
| | **ROVER 5)** | **5.30** | **5.61** | **6.30** | **11.16** | **7.76** | **14.95** | **8.51** | **23.79** | **23.60** | **23.70** |
| 8ch | CSP+BF+derev. | 10.94 | 11.69 | 10.98 | 16.33 | 12.79 | 21.39 | 14.02 | 34.33 | 36.93 | 35.63 |
| | +NLMS | 10.94 | 12.32 | 11.38 | 17.59 | 13.46 | 22.96 | 14.78 | 35.32 | 35.28 | 35.30 |
| | GMM (f-bMMI) | 6.57 | 6.93 | 6.80 | 9.93 | 7.47 | 12.76 | 8.41 | 20.22 | 23.19 | 21.71 |
| | GMM (SAT, f-bMMI) | 6.17 | 6.64 | 6.51 | 10.13 | 7.40 | 13.15 | 8.33 | 20.63 | 23.67 | 22.15 |
| | SGMM (SAT, bMMI) | 5.86 | 6.44 | 6.29 | 9.23 | 6.96 | 12.83 | 7.94 | 20.66 | 23.50 | 22.08 |
| | DNN (SAT, bMMI) | 5.64 | 6.18 | 6.16 | 9.29 | 7.08 | 12.40 | **7.79** | 19.35 | 22.28 | **20.82** |
| | **ROVER 5)** | **4.96** | 5.62 | **5.58** | 8.18 | **5.73** | **10.47** | **6.76** | **16.90** | **20.29** | **18.60** |
| | *ROVER 6)* | 5.00 | **5.56** | **5.38** | **8.15** | **5.73** | 10.70 | *6.75* | 17.47 | 20.36 | 18.93 |

#### 5.5.6.6 Black-box optimization

For eight-channel data, black-box optimization was performed. Fig. 5.7 shows the average WER in terms of the iteration number. WER almost decreased monotonically and, after 100 iterations, it converged. Among these iterations, the results that achieved the best WER on average, are shown in the last column of Table 5.25. The performance improved mainly for the RealData.

#### 5.5.6.7 Evaluation set

Table 5.26 shows the results for the evaluation set (**eva**). Legend of the table is the same to the development set. The optimal system combination is determined based on the WER on the development set. The discriminative training of acoustic model (bMMI) and feature-space discriminative training (f-bMMI) significantly improved the performance. SGMM was better than GMM because model adaptation was well performed. DNN outperformed GMM and SGMM. The DNN with discriminative training achieved the best performance for the SimData and RealData among single systems. This shows the robustness of DNN in unseen conditions. Moreover, system combination [ROVER 5)] improved the WER by 1.0–1.3% for the SimData and 2.1–2.2% for the RealData, respectively. Among system combination systems, the performance of ROVER 5) was better than that of ROVER 6), which used blackbox optimization and could be overly tuned on the development set.

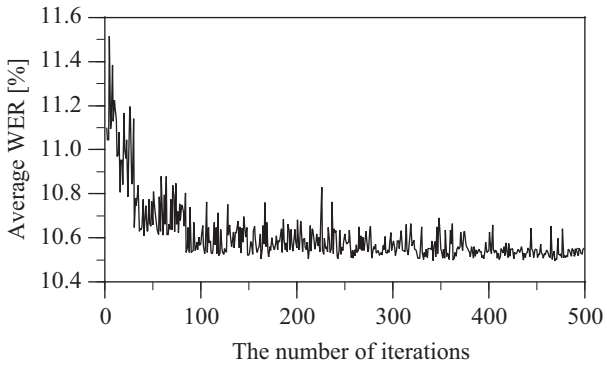



Fig. 5.7 WER [%] averaged over SimData and RealData through black-box optimization of the system selection and parameter setting for ROVER in terms of the number of iterations.

### 5.5.7 Conclusion

We evaluated the medium-sized vocabulary continuous speech recognition task of the REVERB challenge in order to validate the effectiveness of single-channel dereverberation and multi-channel beamforming techniques and discriminaive training of acoustic model and feature transformation in reverberant environments. For speech enhancement, experiments shows the effectiveness of dereverberation of the late reverberation components, and beamforming using multiple microphones that enhances direct sounds compared to the reflected sounds.

For speech recognition, we validated the effectiveness of feature transformations and discriminative training. Experiments show that these techniques are effective across various types of reverberation as well as in noisy environments. To improve robustness in eight types of environments, the system combination approach was used. From two to sixteen systems were constructed to address the problem where the best performing system was different from environment to environment. System combination improved performance; in almost all cases, the combined system outperformed the best performing single system. Our proposed method to specifically provide desired complementary systems for system combination further improved performance. The best results were submitted to the REVERB challenge workshop, and our results were the best among the challenge participants in the same category, which clarifies the effectiveness of our proposed approach.

## 5.6 Source localization and VAD in house (The DIRHA challenge)

The Distant-speech Interaction for Robust Home Applications (DIRHA) project [302] tackles the problem of distant speech interaction in home environments using multiple microphones. A challenge was derived from this project, comprising two major tasks: speaker localization and VAD.

For speaker localization, speakers must be localized in 2D or 3D. It is fairly easy to determine the speaker direction only (1D). For example, the CSP method (Section 2.2.3) is effective even under noisy environments. However, 2D speaker localization is much harder than direction estimation, because it is susceptible to errors, but it is also more attractive. We propose a template-based method in Section 2.4.

For VAD, statistical methods [47, 303] have achieved great success. These methods are robust to noise. However, one difficulty of this challenge is that there are five rooms and the utterances from other rooms must be rejected. Speech detectors can discriminate speech from noise but cannot easily discriminate between speech from the target room and speech from other rooms. To address this problem, integration of speaker localization and VAD is necessary. We propose to utilize speaker localization results for speech detection through the use of either a minimum cost criterion or a classifier-based strategy.

This section mainly proposes an integration method of speaker localization (Section 2.4) and VAD (Sections 3.5 and 5.6.2) is described in Section 5.6.3. Experiments show that the proposed template-based method improves the localization performance and that our classifier-based strategy improves VAD performance.

### 5.6.1 System overview

Fig. 5.8 shows a schematic diagram of the proposed system, which consists of a speaker localization part and a speech detection part. For the speaker localization part, $M$ input pairs are selected from $N$ microphone inputs and the corresponding $M$ TDOAs $\tau$ are calculated by the CSP method. Comparing these TDOAs with the theoretical TDOAs, the 2D-CSP method outputs localized coordinates $\mathbf{s}$ with costs $P(\mathbf{s})$, and the template-based method compensates for errors using reference TDOAs. For the speech detection part, likelihood ratio approaches are adopted. Here, Sohn's method (Section 3.5) and a switching Kalman filter based method (Section 5.6.2) are used. Detections are done per microphone input, and the $N$ detection results are combined using majority voting. In the real data, there are system replies between utterances. These replies are detected separately, and the corresponding utterances are deleted if they exist in the above detection results. Finally, the detection results are modified using a minimum cost criterion or a classifier-based strategy which combine costs $P$ and average powers in each room.

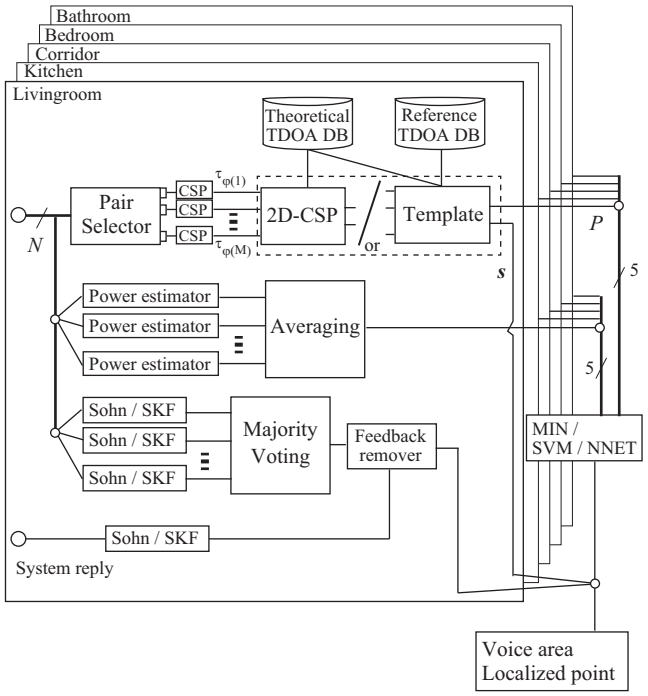



Fig. 5.8 Schematic diagram of the proposed system for the "Living room" localization and detection. (CSP: cross spectrum phase analysis, TDOA: time difference of arrival, Sohn: Sohn's speech detection, SKF: switching Kalman filter based speech detection, MIN: minimum cost criterion, SVM: support vector machine, NNET: neural network)

### 5.6.2 Switching-Kalman-filter-based VAD

The state-of-the-art switching Kalman filter based speech detection method [303] builds the noisy speech model frame by frame, from a prepared clean speech model and a noise model which is estimated online. The features considered are the $K_Y$-dimensional log-Mel spectra $\boldsymbol{Y} = \{Y_k\}_{k=1}^{K_Y}$. In the log-Mel domain, the observed features of speech can be represented as a logarithmic summation of those of clean speech and noise. The likelihoods under the noisy speech and the noise models are each given through a GMM whose components are updated by switching Kalman filters. The likelihood ratio calculation is performed in the same way as in Eqs (3.30) and (3.31), replacing the Gaussians on $X_k$ by the GMMs on $Y_k$.

### 5.6.3 Ensemble integration of calibrated speaker localization and statistical VAD

In this challenge, the utterances from other rooms must be rejected. We propose to use localization results in the other rooms to do so.

#### 5.6.3.1 Minimum cost criterion

Our first approach is to compare the localization cost $P$ in the target room $P_{in}$ with those in the other rooms $P_{out}$. If a speaker is localized in multiple rooms, selecting the speaker location which results in the minimum cost across rooms appears to be the most reasonable. However, simple comparisons lead to many false rejections, because the cost features are dependent on the room shape and microphone settings and thus cannot be simply compared. We thus introduce a tolerance parameter $\eta'$, and for each frame, set a flag $f$ indicating whether the frame's cost is close to being the smallest among all rooms:

$$f = \begin{cases} \text{true} & (\forall P_{out}, P_{in} < \eta' P_{out}) \\ \text{false} & (\text{otherwise}) \end{cases}$$

For each utterance, if the ratio of the number of true flags to the total number of frames is under some thresholds, the utterance is rejected.

#### 5.6.3.2 Classifier-based strategy

In a second approach, we use a classifier $\mathcal{C}$ whose input is a concatenated vector of features from the target room $\mathbf{z}_{in}$ and features from the other rooms $\mathbf{z}_{out}$. After training the classifier on the development set, the classifier outputs are compared with a threshold $\eta''$ to estimate flags for utterance and each frame, as:

$$f = \begin{cases} \text{true} & (\mathcal{C}([\mathbf{z}_{in}; \mathbf{z}_{out}]) > \eta'') \\ \text{false} & (\text{otherwise}) \end{cases}$$

These flags are then combined as in 5.6.3.1 to determine whether to reject the utterance.

### 5.6.4 Experimental setups

Fig. 5.9 shows the setup of the experiment. To simulate voice-active home appliances, synchronously recorded sound files (approximately 1-2 min) were provided by the DIRHA consortium [273]. To simulate realistic environments, these databases were recorded in a real house, which consisted of five rooms: a kitchen, living room, corridor, bathroom and bedroom. Localization was limited to the kitchen and living room and, for these rooms, a circular six-microphone array was installed at the center of the room. Additionally, for all rooms, several two- or three-microphone arrays were installed on the walls encompassing the room. In total, 40 microphones were used. Microphone pairs were selected within each array because microphones belonging to separate arrays were far apart and their correlations were too small.

A development set (**dev**) and a test set (**test**) were provided. According to the regulations, any parameters in the **dev** set can be tuned. Both sets consist of Real and Simulations subsets. In the Real set, for each task, there is only one speaker in one room, who moves around the room. To simulate the dialog between the speaker and system, the replies of the

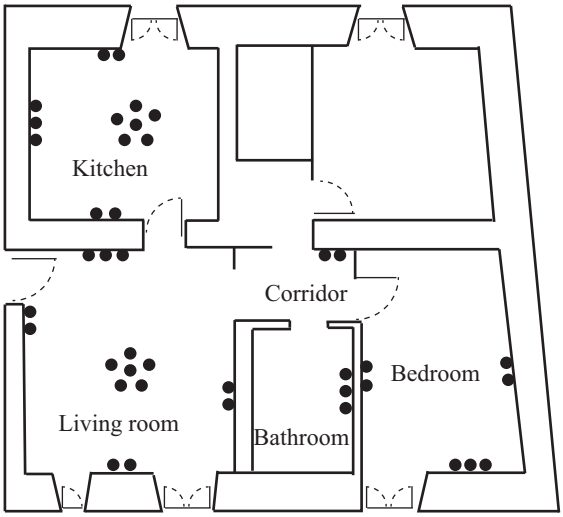



Fig. 5.9 Setting of a house and microphone arrays.

system sometimes break in, but they are provided separately. In the SIMULATIONS set, there can be multiple speakers in different rooms but they are still. The system performance was evaluated using the provided evaluation tools.

#### 5.6.4.1 Localization

We focused on 2-D localization[12] because height localization is less important than horizontal localization as mentioned in the introduction. The speech data were downsampled from the original 48 kHz to 16 kHz for our experiments. The frame size was 960 and the frame shift was 800. We compared the performances of the 2D-CSP and 2D-CSP+template methods with those of the M-CSP[40] and the SRP-PHAT[13,14][304] methods. Fine errors were defined as localization errors of less than 50 cm. These tasks assume that the source position and voice activity area need to be simultaneously estimated. However, for focusing on the comparison of sound localization, in this case, the correct speech area was given.

#### 5.6.4.2 VAD

The VAD performance was evaluated per utterance in terms of precision, recall, and F value. The frame size was 960 and the frame shift was 160 (with 16 kHz sampling). The maximum silence duration in utterances and minimum duration of utterances were set to 500 ms and 300 ms, respectively. For SKF, the number of Gaussian mixture components was 32, and 20-dimensional Mel-spectra were used. HMM hangover scheme [47] was used for both methods.

[12]The `-2D` option was used for the evaluation tool.
[13]http://www.lems.brown.edu/array/tools/srplems.m
[14]A long frame size (1 second) was used.

Table 5.27 Localization and speech detection results on the development set (**dev**). Methods are indicated for speech activity detection (SAD), source localization (LOC), and their integration (INT). Performance criteria for source localization are Fine Error (FE), Gross Error (GE), and Percentage of Correct localization (PCor). For speech detection, utterance-based criteria are used: Precision (P), Recall (Re), and F value.

| Methods | | | Real | | | | | | Simulations | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SAD | LOC | INT | FE | GE | PCor | P | Re | F | FE | GE | PCor | P | Re | F |
| Oracle | 2D-CSP | - | 298 | 602 | .685 | - | - | - | 309 | 925 | .504 | - | - | - |
| | Template | | 303 | 592 | .719 | - | - | - | 160 | 864 | .643 | - | - | - |
| | M-CSP | | 347 | 1307 | .177 | - | - | - | 348 | 1433 | .208 | - | - | - |
| | SRP-PHAT | | *289* | 826 | .537 | - | - | - | 248 | 987 | .509 | - | - | - |
| Sohn | 2D-CSP | - | 295 | 565 | .709 | .693 | **.957** | .804 | 308 | 836 | .525 | .354 | **.905** | .509 |
| | Template | - | 301 | 537 | .746 | .693 | **.957** | .804 | 161 | 769 | .657 | .354 | **.905** | .509 |
| | | MIN | 301 | 537 | .748 | .744 | **.957** | **.837** | 161 | 769 | .657 | .354 | **.905** | .509 |
| | | SVM | 304 | 528 | .757 | .740 | .826 | .781 | 159 | 749 | .681 | .670 | .836 | .744 |
| | | NNET | 299 | **498** | **.779** | .797 | .826 | .811 | 151 | 732 | .685 | **.800** | .693 | .743 |
| SKF | 2D-CSP | - | 300 | 559 | .699 | .697 | .812 | .750 | 303 | 798 | .548 | .416 | .894 | .568 |
| | Template | - | 306 | 532 | .744 | .697 | .812 | .750 | 158 | 714 | .678 | .416 | .894 | .568 |
| | | MIN | 306 | 528 | .752 | .699 | .768 | .732 | 158 | 709 | .679 | .414 | .889 | .565 |
| | | SVM | 310 | 535 | .741 | .823 | .783 | .802 | 157 | 688 | .699 | .661 | .841 | .740 |
| | | NNET | **292** | 503 | .756 | **.837** | .609 | .705 | **149** | **663** | **.704** | .733 | .778 | **.755** |

After performing speech detection per file, majority voting was used to obtain the final VAD results per room.

#### 5.6.4.3 Integration

Localization costs $P$ and segmental speech powers averaged over microphones in each room were used as the features $\mathbf{z}_{in}$ and $\mathbf{z}_{out}$. For the classifier-based strategy, we used SVM-light (v.6.02)[15] for support vector machine (SVM) based classification (linear SVM) and pyBrain (v.0.31)[16] for neural network (NNET) based classification, after normalizing the features to have unit variance. SVM and NNET were trained using binary outputs indicating whether the source was in the target room or not. Parameters and thresholds for SVM and NNET were tuned using the **dev** set. For NNET, the number of hidden layers was two and the number of nodes in the hidden layers was 15 and 10 from the bottom. Finally, for Real, the speech powers of the detected utterances in Livingroom and Kitchen were compared and only the highest one was used because there can be an active speaker only in one room.

Table 5.28 Localization and speech detection results on the test set (**test**).

| Methods | | | Real | | | | | | Simulations | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SAD | LOC | INT | FE | GE | PCor | P | Re | F | FE | GE | PCor | P | Re | F |
| Oracle | 2D-CSP | - | 301 | 622 | .582 | - | - | - | 302 | 1076 | .461 | - | - | - |
| | Template | | 297 | 584 | .658 | - | - | - | 186 | 1094 | .564 | - | - | - |
| Sohn | 2D-CSP | - | 298 | 585 | .610 | .868 | .962 | .913 | 303 | 1004 | .479 | .368 | **.944** | .530 |
| | Template | - | 293 | 550 | .673 | .868 | **.962** | .913 | 185 | 969 | .590 | .368 | **.944** | .530 |
| | | MIN | 293 | 545 | .677 | .882 | **.962** | **.920** | 186 | 970 | .591 | .365 | .934 | .525 |
| | | SVM | 299 | **505** | .678 | **.917** | .316 | .470 | 185 | 961 | .592 | .678 | .939 | **.788** |
| | | NNET | 287 | 542 | .657 | .900 | .532 | .668 | **178** | 969 | .567 | **.720** | .707 | .714 |
| SKF | 2D-CSP | - | 296 | 846 | .624 | .657 | .937 | .772 | 304 | 922 | .526 | .411 | .859 | .556 |
| | Template | - | 292 | 513 | .683 | .657 | .937 | .772 | 184 | 859 | .637 | .411 | .859 | .556 |
| | | MIN | 292 | 512 | **.684** | .651 | .937 | .768 | 184 | 857 | .639 | .411 | .843 | .553 |
| | | SVM | 299 | 518 | .668 | .571 | .367 | .447 | 180 | 838 | .644 | .684 | .813 | .734 |
| | | NNET | **284** | 507 | .662 | .692 | .608 | .647 | 187 | **768** | **.667** | .712 | .742 | .727 |

Table 5.29 Average localization and speech detection results.

| Methods | | | Average (**dev**) | | | | | | Average (**test**) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SAD | LOC | INT | FE | GE | PCor | P | Re | F | FE | GE | PCor | P | Re | |
| Oracle | 2D-CSP | | 306 | 870 | .540 | - | - | - | 302 | 965 | .497 | - | - | - |
| | Template | | 200 | 817 | .658 | - | - | - | 228 | 972 | .592 | - | - | - |
| Sohn | 2D-CSP | - | 305 | 794 | .559 | .414 | **.919** | .570 | 302 | 904 | .517 | .441 | **.949** | .602 |
| | Template | - | 197 | 732 | .673 | .414 | **.919** | .570 | 225 | 870 | .613 | .441 | **.949** | .602 |
| | | MIN | 197 | 732 | .673 | .419 | **.919** | .575 | 225 | 868 | .616 | .441 | .942 | .600 |
| | | SVM | 197 | 714 | .695 | .689 | .833 | .754 | 204 | 920 | .602 | .700 | .762 | **.730** |
| | | NNET | 193 | 692 | .704 | **.799** | .729 | **.762** | 211 | 889 | .588 | **.755** | .657 | .703 |
| SKF | 2D-CSP | - | 302 | 762 | .574 | .461 | .872 | .603 | 301 | 823 | .557 | .462 | .881 | .606 |
| | Template | - | 194 | 686 | .689 | .461 | .872 | .603 | 225 | 768 | .651 | .462 | .881 | .606 |
| | | MIN | 194 | 682 | .692 | .457 | .857 | .596 | 225 | 766 | .653 | .461 | .870 | .602 |
| | | SVM | 196 | 663 | .707 | .694 | .826 | .754 | **203** | 798 | .647 | .664 | .686 | .675 |
| | | NNET | **180** | **642** | **.712** | .753 | .733 | .743 | 215 | **710** | **.666** | .707 | .704 | .706 |

## 5.6.5 Results and discussion

### 5.6.5.1 Localization accuracy with oracle speech detection

To compare the localization accuracies among the above-mentioned methods, the first parts of Tables 5.27 and 5.28 show the results for oracle speech detection cases. Table 5.29 is the average of them. The performance of the 2D-CSP method was higher than those of the multi-channel CSP and SRP-PHAT method. Moreover, the computational complexity was much smaller than those of the multi-channel CSP and SRP-PHAT method. We thus adopted the 2D-CSP method as a baseline. The performance of the template-based method was better than that of the 2D-CSP method significantly, proving effective for the localization in domestic environments.

[15]http://svmlight.joachims.org/
[16]http://pybrain.org/

#### 5.6.5.2 Speech detection accuracy

The second and third parts of Tables 5.27 and 5.28 show the results with speech detection. The performance of SKF was slightly higher than that of Sohn's method. However, neither method by itself was very effective in rejecting noises or leaked utterances from the other rooms. Integration with localization proved effective, but only for the classifier-based strategy. As the classifiers are trained on **dev** data, we compare the results on the **test** set. The performance of the minimum cost criterion was equivalent to that of the baseline. SVM significantly improved the F value, especially with Sohn's method, while NNET improved the F value more consistently with Sohn's method and SKF.

### 5.6.6 Conclusion

We have introduced an effective template-based method that can compensate the discrepancy between the simple spherical wave assumption and the observations, and showed its effectiveness for real domestic environments. In addition, to reject utterances that cannot be easily rejected only by speech detection, we proposed to integrate speaker localization and speech detection. Doing so using classifiers such as SVMs and neural networks improved the speech detection performance.

## 5.7 Conclusion of the chapter

This chapter validated the effectiveness of the proposed method in Chapters 2, 3, and 4. Noisy ASR tasks were CHiME2 in Section 5.2, CHiME3 in Section 5.3, and CHiME4 in Section 5.4. Experiments show that SE methods and discriminative methods were effective. Reverberant ASR task was REVERB challenge in Section 5.5. The proposed dereverberation and discriminative system combinations were effective. Among them, in CHiME2 and REVERB challenge, our team achieved the best results. DIRHA challenge in Section 5.6 shows that combination of localization and VAD is important.

Journal papers related to this chapter are [305, 306] and conference papers are [218, 183, 280, 307, 93, 308].

# 6 Conclusion

## 6.1 Findings of each chapter

This paper aims to improve the ASR performance under noisy and reverberant environments in order to widen the application of ASR. Chapters 2–4 propose various methods including front-end and back-end techniques. Chapter 5 validates the effectiveness of the proposed method in various conditions. Findings of each chapter are as follows.

Chapter 2 describes the details of estimation methods of source localization and direction. Section 2.2 introduces conventional methods and Section 2.3 proposes an effective prior distribution of source direction based on the VAD information to improve the estimation accuracy of source direction. Section 2.4 proposes a template-based method that compensates the discrepancy of TDOA from theoretical TDOA due to location errors of microphones and reverberation.

Chapter 3 describes the details of speech enhancement methods that are important as a preprocess of ASR. Section 3.2 proposes a single-channel dereverberation method that eliminates reverberation based on the estimated reverberation time that represents the extent of reverberation. Section 3.3 proposes a multi-channel method that combines binary masking based on TDOA with IVA. Section 3.4 proposes an effective initialization method that makes the performance of MNMF stable, because the SE performance of MNMF is high performance but is heavily dependent on initial values. Section 3.5 describes a VAD method that detects speech activation from noisy speech. Although conventional method uses two models (speech and noise models), the proposed method uses one density models based on density ratio estimation. Section 3.6 investigates the influence of clipping due to inappropriate recording levels on the ASR performance. Section 3.7 investigates the influence of sampling frequency mismatch between training and evaluation data of on the ASR performance and also proposes to reduce performance drop by using DNN and widen the speech band of evaluation data.

Chapter 4 describes the details of important back-end techniques that are important to realize robust ASR, especially focusing on discriminative methods. Discriminative method is a retraining method that corrects ASR errors by modifying various ASR models. Section 4.2 overviews ASR systems and Section 4.2.3 describes features that are used for ASR. Section 4.3.2 introduces a discriminative method to linear discriminant analysis of acoustic features. Section 4.4.2 describes a discriminative method of acoustic model that is important model for ASR. Section 4.5.2 proposes to combine model size reduction of acoustic models with discriminative method. Section 4.6 proposes a framework that constructs complementary systems that improve the ASR performance when multiple systems are combined by extending an objective function of discrim-

inative method. Section 4.7 proposes a discriminative method of language models that are used for ASR. Section 4.8 reduces the influence of speech distortions that are mixed when speech enhancement (e.g., Chapter 3) is performed.

Chapter 5 validates the effectiveness of the proposed methods on various challenges whose tasks are noise and reverberation robust ASR. Section 5.2 validates the effectiveness of discriminative methods (Section 4.4.2) on noisy ASR in home environments (CHiME2). Sections 5.3 and 5.4 aims to improve the ASR performance in public spaces and validates the performance on CHiME3 and CHiME4 as a benchmark. CHiME3 has a large variety of noise environments and the best suitable system is different from environment to environment. We propose a best ASR system selection based on i-vector features that are used for speaker recognition. CHiME4 ((5.4)) uses state-of-the-art SE methods and various features and systems. The number of ASR errors becomes half of that of the same task in the CHiME3. Section 5.5 aims to improve the performance in various reverberant environments on REVERB challenge. The effectiveness of the proposed dereverberation method ((3.2)) and system combination ((4.6)) was confirmed. Among these challenges, for CHiME2 and REVERB challenge, our team achieved the best performance. By participating these challenges, the effectiveness of the proposed method is evaluated on benchmark tasks. Section 5.6 aims to improve the performance of source localization and VAD. Both performances can be improved by combining the costs of source localization ((2.4)) and VAD ((3.5)).

## 6.2 Future work

As mentioned in the "remaining tasks" of the introduction, future work will realize natural ASR interfaces in addition to improving the robustness of the ASR.

## 6.3 Thesis summary in Japanese

本論文では音声認識の適用先拡大のため、騒音・残響がある環境で音声認識性能を向上させることを目的としている。実環境で認識性能が低下する原因としては、騒音と残響がある。音声認識を行うためには、騒音を除去して音声を強調する技術が必要である。そのためには、複数マイクを使って話者の位置を特定し、マイク間の位相差を手がかりに、音声を強調する。第 2 章では、従来法を紹介したのちに、音声区間の情報に基づき、音源の方向に関する事前分布を形成し、音源方向の推定精度を向上させる方法を提案した。また、マイクロフォンの配置誤差・残響による理論的なマイク間の到来時間差からのずれを、補正する方法を提案した。3 章では、音声認識の前処理として重要な音声強調手法に関して詳述した。1 マイクを使う方法として、残響の程度を表す残響時間を自動的に推定し、残響を除去する方法を提案した。一方、複数のマイクを使う方法として、到来方向に基づくバイナリマスクと独立ベクトル分析を統合する方法を提案した。加えて音声強調法として有効だが初期値依存性が高いことで知られるマルチチャンネル非負値行列因子分解に対して、性能を安定させる初期値を与える方法を示した。これらの成果により、音声から騒音と残響を取り除くことができるようになり、音声認識の性能が大幅に向上することを、実験的に確かめた。音声強調の次に重要なのは、正確に音声区間を切り出してくる技術である。音声区間を取り損なえば音声認識をすることができず、騒音区間が多く混入してもこれもまた音声認識性能を下げる原因となるためである。ここでは、音声区間検出のモデルパラメータ推定に密度比推定を使うことで、従来、音声と騒音の 2 つのモデルが必要だったところを 1 つの密度比モデルに減らすことができ、頑健性が向上した。また学習環境と評価環境でミスマッチがあると、性能が低下する。本論文では、録音レベルが不適切な場合でクリッピングが起こった場合と、サンプリング周波数が学習データと評価データで異なる場合に、どの程度音声認識の性能が低下するかを検討した。サンプリング周波数のずれに関しては、深層神経回路網を用いて評価データの音声の帯域拡張を行うことで、音声認識の性能低下を抑えられる方法を提案した。4 章では、音声認識を騒音・残響に対して頑健にする方法に関して述べている。2 章・3 章の手法により、騒音を抑圧したとはいえ、やはり騒音の影響は残っているため、騒音に頑健な音声認識システムを構築する必要がある。この際に有効なのは、音声認識の誤り率に関連する目的関数を最小化する誤り訂正学習の一種である識別学習である。これにより、音声認識の誤りを訂正するように種々のモデルを再学習する。本論文では、特徴量の線形判別分析、音声認識のモデルとして主要な 2 つのモデルである音響モデルと言語モデルに識別学習を導入し、騒音環境下で音声認識性能が向上することを確認した。併せて、音響モデルのモデル削減法と識別学習を組み合わせる方法を提案した。また、識別学習の目的関数を拡張し、複数のシステムを組み合わせた際に有効である、元のシステムと異なる仮説を出力するような補助システムを構築する枠組みを示した。3 章などで述べた音声強調は、大抵の場合音声認識性能を向上させるが、音声強調を行ったことで音声にひずみが混入し特徴量の分布が変わってしまい、性能が低下してしまうことがある。これに対処するため、起こりうるひずみを想定して学習することで、ひずみによる影響を低減する方法を提案した。最終章である 5 章では、騒音・残響に頑健な音声認識タスクを対象とする様々なチャレンジにより、提案手法の有効性を確認した。主に家庭環境での騒音を対象とした第 2 回 CHiME チャレンジでは、識別的手法の有効性を示した。主に屋外環境の騒音を対象とした第 3 回 CHiME チャレンジでは、騒音環境が大きく異なるため複数のシステムを用意すると最良のシステムが環境ごとに異なる。この中から最良のシステムを話者認識に用いる

i-vector 特徴量を用いて選択する手法を提案した。第 4 回 CHiME チャレンジは第 3 回の翌年のリピートであるが、種々の特徴量・システム、先端的な音声強調手法を用いて、第 3 回 CHiME チャレンジと同一のタスクで音声認識誤りを半減した。種々の残響環境を対象とした REVERB チャレンジでは、上述の残響除去法と識別学習を用いたシステム統合法の有効性を確認した。このうち、第 2 回 CHiME チャレンジ、REVERB チャレンジでは、参加者中トップの成績を収め、提案法の有効性が確認された。DIRHA チャレンジは、音源位置推定と音声区間検出を同時に行うタスクである。ここでは、音源位置推定法と音声区間検出のコストを統合することで両者の性能を向上させられることが分かった。このように本論文では、騒音・残響がある環境で音声認識性能を向上させることを目的として、音源の位置推定法、音声強調法 (騒音抑圧・残響除去)、音声認識の各種モデルの頑健性を向上させる識別的手法を提案し、公開チャレンジにより提案法の有効性を確認した。

# Acknowledgement


## English version

This thesis is a summary of the research, which I had studied in Mitsubishi Electric Corporation. First, I thank Prof. Takao Kobayashi for his support. In addition, I thank co-reviewers for their comments. Second, I thank researchers in Mitsubishi Electric Corporation, especially, Mr. Toshiyuki Hanazawa for his continuous help. Chapter 2 and 3 are collaborative works with Mr. Tomohiro Narita and Prof. Kenichi Furuya. Chapter 4 and 5 are based on the development with Mitsubishi Electric Research Laboratories (MERL). I had stayed for four months in winter Boston and had a close relationship with researchers in MERL. I thank Dr. Shinji Watanabe, Jonathan Le Roux, and John Hershey for their help. Third, I thank colleagues in Denso IT Laboratory, which I belong to now. Finally, I thank my family members for their supports.

## Japanese version

本論文は筆者が三菱電機株式会社 情報技術総合研究所 音声言語処理技術部において行った研究をまとめたものです。主査の小林隆夫教授には、三菱電機での共同研究をきっかけとして、学会で会うたびに気にかけていただき、今回の学位取得につながりました。副査の先生方にも、数々の審査の際にはお世話になりました。御礼申し上げます。

建築音響の分野で修士課程を修了し、三菱電機に入社以来、10 年間音声言語処理に関する研究開発を行ってきましたが、入社までは音声処理は全く経験がありませんでした。そのような筆者に対して、新入社員の教育担当として、音声認識の基礎から教えてくださった花沢利行主任研究員にまず感謝したいと思います。音声認識の草創期から研究に携われ、IEEE 論文賞も受賞されるなど輝かしい経歴を持つ花沢研究員の妥協を許さない研究姿勢には、学ぶべきところがたくさんありました。入社のきっかけをくださり、熱心にご指導いただいた岩崎知弘元音声チームリーダー、高橋真哉元音声言語処理技術部長にも感謝します。本論文の 2,3 章の信号処理技術は、成田知宏主任研究員と一緒に開発したものです。成田研究員には、信号処理のイロハなどをご教授いただき、本論文で実装した音源定位のアルゴリズムは成田研究員の実装をもとにしております。3 章のマルチチャンネル NMF の開発は、大分大学古家賢一教授との共同研究によっています。この共同研究では、インターン生の受け入れ等、ありがたい経験をさせていただきました。

4,5 章の技術開発は Mitsubishi Electric Research Laboratories (MERL) との共同研究が基礎となっています。筆者は 2012 年末から 2013 年にかけての冬の時期にボストンに 4ヶ月ほど滞在し、密な関係を築けたことが本成果につながりました。MERL で暖かく受け入れてくださった渡部晋治元研究員、ジョナトンルルー研究員、ジョンハーシー元マネージャーに感謝したいと思います。

MERL 研究員の方々には、国際会議に通っていなかった筆者に、研究の進め方、論文の書き方等の具体的なアドバイスをいただき、力をつけることができました。ICASSP/ASRU/INTERSPEECH 等の音声認識の分野でトップ会議の論文は、ほぼ MERL との共著となっており、この論文の礎を共に築いてくださったことに感謝いたします。石井純元音声グループマネージャー (現知識部長) には、研究活動を進める財政面でサポートいただきました。音声言語処理技術部唯一の同期であった杉原堅也さんとは、研修論文など大変なイベントを共に乗り切りました。後輩の粟野智治さん、新山摩梨花さん、細谷耕佑さんと朝の勉強会などを行ったのもとても楽しい思い出です。音声グループの後輩であった金川裕紀さんとは、ともに研究活動を進めるとともに、プライベートな部分でも楽しく過ごすことができました。ありがとうございました。

さて、今回学位取得を思い立ったのは、三菱電機 故片木顧問による喜連川先生の話 [309] がきっかけになっています。筆者の勤めていた情報技術総合研究所では、毎年度末にその年に学位を取得した人による学位論文発表会が行われています。学位取得の裏話や苦労譚など貴重な話が聞けるため、筆者はこのイベントがとても好きで、特段の事情がない限り毎年出席しておりました。情報技術総合研究所の特に光電波部門の研究者たちはアカデミアでも認められており、毎年多数の学位取得者を輩出しておりました。そこで片木顧問が毎年最後に、光電波部門の人たちが研究できるのは喜連川さんがそのような道を切り開いたからだというお話をされていました。アンテナ技術者としての片木顧問の研究姿勢には感銘を受け、私淑しておりました。筆者の所属していたマルチメディア部門は光電波部門ほどアカデミックな雰囲気ではありませんでしたが、筆者は光電波部門を目標とし、三菱電機のアカデミアでの認知度を上げたいと思い研究活動を続けてきました。残念ながら目標であった学位論文発表会での発表は筆者の退職に伴い叶いませんでしたが、きっかけをいただいたことにとても感謝しています。自由に研究活動を進めさせていただいた三菱電機には、感謝の念にたえません。

転職先である株式会社 デンソーアイティーラボラトリの皆様にも、ゼミや審査会等への出席に便宜を図っていただき、感謝しております。これからは培った音声認識の技術を応用して、対話を含む自然言語の理解に関する研究を進めていきたいと思います。

また本論文の成果とは直接の関係はありませんが、著者の研究生活の基礎を築いてくださった修士課程の指導教官である東京大学大学院 新領域創成科学研究科 社会文化環境学専攻 佐久間哲哉准教授および神奈川大学工学部 建築学科 安田洋介教授にも感謝します。先生方のおかげで、筆者の最初の論文 [310] が受理されたことで、論文執筆のやり方を習得することができました。また佐久間研究室 客員共同研究員の李孝振さん、江田和司さん、同特任研究員の井上尚久さんとは学会での交流が励みになりました。

最後になりましたが、本研究を暖かく見守ってくれた家族・親戚に感謝したいと思います。

# References


[1] F.d. Saussure, Cours de linguistique générale, Arbre d'Or, 2005 (1st 1916).

[2] 時枝誠記，国語学原論，岩波書店，1941.

[3] 金田一春彦，日本語，岩波書店，1988.

[4] F. Jelinek, "Continuous speech recognition by statistical methods," Proceedings of IEEE, vol.64, pp.532–556, 1976.

[5] S.B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE Transactions on Acoustics, Speech and Signal Processing, vol.ASSP-28-4, pp.357–366, Aug. 1980.

[6] 迫江博昭，千葉成美，"動的計画法を利用した音声の時間正規化に基づく連続音声認識，" 日本音響学会誌，vol.27，no.9，pp.483–490，1971.

[7] H. Sakoe, "Two-level DP matching – a dynamic programming-based pattern matching algorithm for connected word recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.ASSP-27, no.6, pp.588–595, 1979.

[8] G.D. Forney, "The Viterbi algorithm," Proceedings of IEEE, vol.61-3, pp.268–278, March 1973.

[9] L. Rabiner and B. Juang, "An introduction to hidden Markov models," IEEE ASSP Magazine, vol.3, pp.4–16, 1986.

[10] 大河内正明，"Hidden Markov model に基づいた音声認識，" 日本音響学会誌，vol.42，no.12，pp.936–941，1986.

[11] A.B. Poritz, "Hidden Markov model: A guided tour," Proceedings of ICASSP, vol.1, pp.7–13, 1988.

[12] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition: Neural networks vs hidden Markov models," Proceedings of ICASSP, vol.1, pp.107–110, April 1988.

[13] S. Austin, C. Barry, Y.-L. Chow, M. Derr, O. Kimball, F. Kubala, J. Makhoul, P. Placeway, W. Russell, R. Schwartz, and G. Yu, "Improved HMM models for high performance speech recognition," Human Language Technology Conference Proceedings of the workshop on Speech and Natural Language, pp.249–255, 1989.

[14] L.R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of IEEE, vol.77-2, pp.257–286, 1989.

[15] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, “Phoneme recognition using time-delay neural networks,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.31-3, pp.328–339, March 1989.

[16] X. Huang, Y. Ariki, and M. Jack, Hidden Markov Models for speech recognition, Edinburgh university press, 1990.

[17] K. Kita, T. Kawabata, and T. Hanazawa, “HMM continuous speech recognition using stochastic language models,” Proceedings of ICASSP, vol.1, pp.581–584, April 1990.

[18] T. Hanazawa, K. Kita, S. Nakamura, T. Kawabata, and K. Shikano, “ATR HMM-LR continuous speech recognition system,” Proceedings of ICASSP, vol.1, pp.53–56, April 1990.

[19] B. Juang and L. Rabiner, “Hidden Markov models for speech recognition,” TECHNOMETRICS, vol.33-3, pp.251–272, Aug. 1991.

[20] P. Price, W. Fisher, J. Bernstein, and D. Pallett, “The DARPA 1000-word resource management database for continuous speech recognition,” Proceedings of ICASSP, pp.651–654, 1988.

[21] 中川聖一，鹿野清宏，東倉洋一，音声・聴覚と神経回路網モデル，オーム社，1990.

[22] R. Kuhn and R. De Mori, “A cache-based natural language model for speech recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.12, pp.570–583, 1990.

[23] J. Allen, “How do humans process and recognize speech?,” IEEE Transactions on Speech and Audio Processing, vol.2, pp.567–577, Oct. 1994.

[24] M. Boros, W. Eckert, F. Gallwitz, G. Gorz, G. Hanrieder, and H. Niemann, “Towards understanding spontaneous speech: Word accuracy vs. concept accuracy,” Proceedings of International Conference on Spoken Language Processing, pp.1009–1012, 1996.

[25] C. Wooters and M. Huijbregts, “The ICSI RT07s speaker diarization system,” Multimodal Technologies for Perception of Humans, pp.509–519, Springer, 2008.

[26] M. Delcroix, K. Kinoshita, T. Nakatani, S. Araki, A. Ogawa, T. Hori, S. Watanabe, M. Fujimoto, T. Yoshioka, T. Oba, Y. Kubo, M. Souden, S.-J. Hahm, and A. Nakamura, “Speech recognition in living rooms: Integrated speech enhancement and recognition system based on spatial, spectral and temporal modeling of sounds,” Computer Speech and Language, vol.27, pp.851–873, 2013.

[27] Y. Tachioka, T. Narita, and J. Ishii, “Semi-blind source separation using binary masking and indepedent vector analysis,” IEEJ Transactions on Electrical and Electronic Engineering, vol.10, no.1, pp.114–115, Jan. 2015.

[28] E. Menegatti, E. Mumolo, M. Nolich, and E. Pagello, “A surveillance system based on audio and video sensory agents cooperating with a mobile robot,” Intelligent Autonomous System, vol.8, pp.335–343, 2004.

[29] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.24, no.4, pp.320–327, Aug. 1976.

[30] R. Schmidt, “Multiple emitter location and signal parameter estimation,” IEEE Transactions on Antennas Propagation, vol.34, pp.276–280, March 1986.

[31] M. Omologo and P. Svaizer, “Acoustic event localization using a crosspower-spectrum phase based technique,” Proceedings of ICASSP, vol.II, pp.273–276, 1994.

[32] M. Omologo and P. Svuizer, “Acoustic source location in noisy and reverberant environment using CSP analysis,” Proceedings of ICASSP, vol.2, pp.921–924, 1996.

[33] O. Ichikawa, T. Fukuda, and M. Nishimura, “DOA estimation with local-peak-weighted CSP,” EURASIP Journal on Advances in Signal Processing, 2010.

[34] Y. Denda, T. Nishiura, and Y. Yamashita, “Robust talker direction estimation based on weighted CSP analysis and maximum likelihood estimation,” IEICE Transactions on Information and Systems, vol.E89-D, pp.1050–1057, 2006.

[35] T. Nishiura, T. Yamada, S. Nakamura, and K. Shikano, “Localization of multiple sound sources based on a CSP analysis with a microphone array,” Proceedings of ICASSP, vol.2, pp.1053–1056, 2000.

[36] 西浦敬信，西岡良典，山田武志，中村　哲，鹿野清宏，“CSP 法による音源位置同定を備えたマルチビームフォーミング,” 電子情報通信学会論文誌 D，vol.J83-D2，pp.1610–1619，July 2000.

[37] Y. Tachioka, T. Narita, and T. Iwasaki, “Direction of arrival estimation by cross-power spectrum phase analysis using prior distributions and voice activity detection information,” Acoustical Science & Technology, vol.33, no.1, pp.68–71, Jan. 2012.

[38] K. Nakadai, H.G. Okuno, and H. Kitano, “Real-time sound source localization and separation for robot audition,” Proceedings of the IEEE International Conference on Spoken Language Processing, pp.193–196, 2002.

[39] D.V. Rabinkin, R.J. Renomeron, A. Dahl, J.C. French, J.L. Flanagan, and M.H. Bianchi, “A DSP implementation of source location using microphone arrays,” Proceedings of SPIE, pp.88–99, 1996.

[40] K. Hayashida, M. Morise, and T. Nishiura, "Near field sound source localization based on cross-power spectrum phase analysis with multiple channel microphones," Proceedings of INTERSPEECH, pp.2758–2761, Sept. 2010.

[41] K. Ho and L. Yang, "On the use of a calibration emitter for source localization in the presence of sensor position uncertainty," IEEE Transactions on Signal Processing, vol.56, pp.5758–5772, 2008.

[42] Y. Tachioka and T. Narita, "Template-based method for compensation of time difference of arrival in passive sound source localization under reverberant and noisy environments," Journal of Signal Processing, vol.21, no.2, pp.73–79, March 2017.

[43] R. Kennedy, T. Abhayapala, and D. Ward, "Broadband nearfield beamforming using a radial beampattern transformation," IEEE Transactions on Signal Processing, vol.46, pp.2147–2156, 1998.

[44] T. Suzuki and Y. Kaneda, "Sound source direction estimation based on subband peak-hold processing," The Journal of the Acoustical Society of Japan, vol.65, no.10, pp.513–522, Oct. 2009.

[45] M.S. Brandstein and H.F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," Proceedings of ICASSP, vol.1, pp.375–378, 1997.

[46] J. Chen, J. Benesty, and Y. Huang, "Robust time delay estimation exploiting spatial correlation," Proceedings of ICASSP, vol.5, pp.481–484, 2003.

[47] J. Sohn, N.S. Kim, and W. Sung, "A statistical model-based voice activity detection," IEEE Signal Processing Letters, vol.6, pp.1–3, Jan. 1999.

[48] D.D. Lee and S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol.401, no.6755, pp.788–791, Oct. 1999.

[49] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," IEEE Transactions on Audio, Speech and Language Processing, vol.15, no.1, pp.1–12, Jan. 2007.

[50] K. Kinoshita, T. Nakatani, and M. Miyoshi, "Spectral subtraction steered by multi-step forward linear prediction for single channel speech dereverberation," Proceedings of ICASSP, vol.I, pp.817–820, 2006.

[51] K. Kinoshita, M. Delacroix, T. Nakatani, and M. Miyoshi, "Multi-step linear prediction based speech enhancement in noisy reverberant environment," Proceedings of INTERSPEECH, pp.854–857, 2007.

[52] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Transactions on Acoustics, Speech and Signal Processing, vol.27, no.2, pp.113–120, April 1979.

[53] T. Nakatani and M. Miyoshi, “Blind dereverberation of single channel speech signal based on harmonic structure,” Proceedings of ICASSP, vol.1, pp.92–95, 2003.

[54] K. Kinoshita, T. Nakatani, and M. Miyoshi, “Fast estimation of a precise dereverberation filter based on the harmonic structure of speech,” Acoustical Science & Technology, vol.28, no.2, pp.105–114, 2007.

[55] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, “Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.24, no.9, pp.1626–?1641, 2016.

[56] M. Miyoshi and Y. Kaneda, “Inverse filtering of room acoustics,” IEEE Transactions on Acoustics, Speech and Signal Processing, vol.36-2, pp.145–152, Feb. 1988.

[57] S. Gannot and M. Moonen, “Subspace methods for multi-microphone speech dereverberation,” EURASIP Journal of Applied Signal Process, vol.11, pp.1074–1090, 2003.

[58] A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis, John Wiley & Sons, 2001.

[59] J. Benesty, S. Makino, and J. Chen, Speech Enhancement, Springer, 2005.

[60] P.A. Naylor and N.D. Gaubitch, Speech Dereverberation, Springer, 2010.

[61] K. Lebart, J. Boucher, and P. Denbigh, “A new method based on spectral subtraction for speech dereverberation,” Acta Acoustica, vol.87, pp.359–366, 2001.

[62] R. Gomez, J. Even, H. Saruwatari, and K. Shikano, “Distant-talking robust speech recognition using late reflection components of room impulse response,” Proceedings of ICASSP, pp.4581–4584, 2008.

[63] H. Löllmann and P. Vary, “Low delay noise reduction and dereverberation for hearing aids,” EURASIP Jornal on Advances in Signal Processing, 2009.

[64] O. Yilmaz and S. Rickard, “Blind separation of speech mixtures via time-frequency masking,” IEEE Transactions on Signal Processing, vol.52, pp.1830–1847, 2004.

[65] H. Kuttruff, Room Acoustics, Spon Press, 2000.

[66] W.G. Bradford and L.E. Atlas, “Acoustic diversity for improved speech recognition in reverberant environments,” Proceedings of ICASSP, vol.1, pp.557–600, 2002.

[67] S. Bistafa and J. Bradley, “Predicting reverberation times in a simulated classroom,” Journal of Acoustical Society of America, vol.108, pp.1721–1731, 2000.

[68] I. National Institute of, “Speech resources,” http://research.nii.ac.jp/src/eng/list/.

[69] M. Schroeder, "New method of measuring reverberation time," Journal of Acostical Society of America, vol.37, pp.409–412, 1965.

[70] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," IEEE Transactions on Audio, Speech, and Language Processing, vol.19, pp.516–527, 2011.

[71] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp.189–192, Oct. 2011.

[72] P. Zolfaghari, S. Watanabe, A. Nakamura, and S. Katagiri, "Bayesian modelling of the speech spectrum using mixture of Gaussians," Proceedings of ICASSP, vol.1, pp.553–556, 2004.

[73] J. Cermak, S. Araki, H. Sawada, and S. Makino, "Blind source separation based on a beamformer array and time frequency binary masking," Proceedings of ICASSP, vol.1, pp.145–148, 2007.

[74] D. Kolossa, Independent Component Analysis for Environmentally Robust Speech Recognition, PhD Thesis, Technischen Universität Berlin, 2008.

[75] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Grouping separated frequency components by estimating propagation model parameters in frequency-domain blind source separation," IEEE Transactions on Audio, Speech and Language Processing, vol.15, pp.1592–1604, July 2007.

[76] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," Neurocomputing, vol.41, pp.1–24, 2001.

[77] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithm, pp.1027–1035, 2007.

[78] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization," Pattern Recognition, vol.41, pp.1350–1362, 2008.

[79] H. Qiao, "New SVD based initialization strategy for non-negative matrix factorization," Pattern Recognition Letters, vol.63, pp.71–77, Oct. 2015.

[80] K. Kwon, J.W. Shiny, I. Choi, H.Y. Kim, and N.S. Kim, "Incremental approach to NMF basis estimation for audio source separation," Proceedings of APSIPAAPSIPA, pp.1–5 Dec. 2016.

[81] B. Raj, R. Singh, and T. Virtanen, "Phoneme-dependent NMF for speech enhancement in monaural mixtures," Proceedings of INTERSPEECHISCA, pp.1217–1220 Aug. 2011.

[82] F. Sohrab and H. Erdogan, "Recognize and separate approach for speech denoising using nonnegative matrix factorization," Proceedings of EUSIPCO, Sept. 2015.

[83] K. Kinoshita, M. Delcroix, A. Ogawa, and T. Nakatani, "Text-informed speech enhancement with deep neural networks," Proceedings of INTERSPEECH, pp.1760–1764, 2015.

[84] Z. Wang and D. Wang, "A joint training framework for robust automatic speech recognition," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.24, no.4, pp.796–806, April 2016.

[85] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," IEEE Transactions on Audio, Speech, and Language Processing, vol.18, no.3, pp.550–563, March 2010.

[86] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Multichannel extensions of non-negative matrix factorization with complex-valued data," IEEE Transactions on Audio, Speech, and Language Processing, vol.21, no.5, pp.971–982, May 2013.

[87] I. Miura, Y. Tachioka, T. Narita, J. Ishii, F. Yoshiyama, S. Uenohara, and K. Furuya, "Analysis of initial-value dependency in multichannel nonnegative matrix factorization for blind source separation and speech recognition (in Japanese)," IEICE Transactions on Information and Systems, vol.J100-D, no.3, pp.376–384, March 2017.

[88] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Efficient multichannel nonnegative matrix factorization exploiting rank-1 spatial model," Proceedings of ICASSP, pp.276–280 April 2015.

[89] C. Févotte, N. Bertin, and J. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," Neural Computation MIT Press, vol.21, pp.793–830, 2009.

[90] M. Nakano, H. Kameoka, J. Le Roux, Y. Kitano, N. Ono, and S. Sagayama, "Convergence-guaranteed multiplicative algorithms for non-negative matrix factorization with beta-divergence," Proceedings of MLSP, pp.283–288, 2010.

[91] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," IEEE Transactions on Audio, Speech and Language Processing, vol.15, no.7, pp.2011–2023, 2007.

[92] 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," Proceedings of ASRU, pp.436–443, IEEE, Dec. 2015.

[93] Y. Tachioka, S. Watanabe, and T. Hori, “The MELCO/MERL system combination approach for the fourth CHiME challenge,” Proceedings of the Fourth CHiME Challenge Workshop, pp.1–3, Sept. 2016.

[94] 石塚健太郎，藤本雅清，中谷智広，“音声区間検出技術の最近の研究動向,” 日本音響学会誌，vol.65，pp.537–543，Oct. 2009.

[95] L. Rabiner and M. Sambur, “An algorithm for determining the endpoints of isolated utterances,” The Bell System Technical Journal, vol.54, pp.297–315, Feb. 1975.

[96] K. Ishizuka and T. Nakatani, “Study of noise robust voice activity detection based on periodic component to aperiodic component ratio,” Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition (SAPA), pp.65–70, Sept. 2006.

[97] M. Fujimoto, K. Ishizuka, and T. Nakatani, “A voice activity detection based on the adaptive integration of multiple speech features and a signal decision scheme,” Proceedings of ICASSP, pp.4441–4444, 2008.

[98] T. Ohnish, P. Dixon, K. Iwano, and S. Furui, “Robust speech recognition using VAD-measure embedded decoder,” Proceedings of INTERSPEECH, pp.2239–2242, Sept. 2009.

[99] J. Ramìrez and J. Segura, “Statistical voice activity detection using a multiple observation likelihood ratio test,” IEEE Signal Processing Letter, vol.12, pp.689–692, Oct. 2005.

[100] M. Fujimoto, K. Ishizuka, and H. Kato, “Noise robust voice activity detection based on statistical model and parallel non-linear Kalman filtering,” Proceedings of ICASSP, vol.4, pp.797–800, 2007.

[101] J.H. Chang, N.S. Kim, and S.K. Mitra, “Voice activity detection based on multiple statistical models,” IEEE Transactions on Signal Processing, vol.54, pp.1965–1976, June 2006.

[102] O. Pernìa, J. Górriz, J. Ramìrez, C. Puntonet, and I. Turias, “An efficient VAD based on a generalized Gaussian PDF,” Proceedings of International Conference on Advances in Nonlinear Speech Processing, pp.246–254, 2007.

[103] M. Sugiyama, T. Kanamori, T. Suzuki, S. Hido, J. Sese, I. Takeuchi, and L. Wang, “A density-ratio framework for statistical data processing,” IPSJ Transactions on Computer Vision and Applications, vol.1, pp.183–208, 2009.

[104] 杉山　将，“密度比推定に基づく機械学習の新たなアプローチ,” 統計数理，vol.58，pp.141–155，2010.

[105] Y. Kawahara and M. Sugiyama, “Change-point detection in time-series data by direct density-ratio estimation,” Proceedings of SIAM International Conference on Data Mining, pp.389–400, April 2009.

[106] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz, Nonparametric and Semiparametric Models, Springer Series in Statistics, 2004.

[107] S. Lab, http://sugiyama-www.cs.titech.ac.jp/˜sugi/software/KLIEP/.

[108] S. Rennie, T. Kristjansson, P. Olsen, and R. Gopinath, "Dynamic noise adaptation," Proceedings of ICASSP, vol.I, pp.1197–1200, 2006.

[109] R.J. Weiss and T. Kristjansson, "DySANA: Dynamic speech and noise adaptation for voice activity detection," Proceedings of INTERSPEECH, pp.127–130, 2008.

[110] N. Kitaoka, T. Yamada, S. Tsuge, C. Miyajima, K. Yamamoto, T. Nishiura, Y. Denda, M. Fujimoto, T. Takiguchi, S. Tamura, S. Matsuda, T. Ogawa, S. Kuroiwa, K. Takeda, and S. Nakamura, "CENSREC-1-C: An evaluation framework for voice activity detection under noisy environments," Acoustical Science & Technology, vol.30, pp.363–371, Sept. 2009.

[111] S. Maymon, E. Marcheret, and V. Goel, "Restoration of clipped signals with application to speech recognition," Proceedings of INTERSPEECH, pp.3294–3297, Aug. 2013.

[112] T. Yamada, M. Kumakura, and N. Kitawaki, "Performance estimation of speech recognition system under noise conditions using objective quality measures and artificial voice," IEEE Transactions on Audio, Speech, and Language Processing, vol.14, pp.2006–2013, Nov. 2006.

[113] T. Fukumori, M. Nakayama, T. Nishiura, and Y. Yamashita, "Estimation of speech recognition performance in noisy and reverberant environments using PESQ score and acoustic parameters," Proceedings of APSIPA ASC, Oct. 2013.

[114] ITU-T Rec P862, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end spech quality assessment of narrow band telephone networks and speech codecs," http://www.itu.int/rec/T-REC-P862/, 2001.

[115] S. Gazor and W. Zhang, "Speech probability distribution," IEEE Signal Processing Letter, vol.10, no.7, pp.204–207, July 2003.

[116] A. Lee, T. Kawahara, and K. Shikano, "Julius – an open source real-time large vocabulary recognition engine," Proceedings of EUROSPEECH, pp.1691–1694, 2001.

[117] A. Lee, T. Kawahara, K. Takeda, and K. Shikano, "A new phonetic tied-mixture model for efficient decoding," Proceedings of ICASSP, vol.3, pp.1269–1272, 2000.

[118] Y. Wang, S. Zhao, Y. Yu, and J. Kuang, "Speech bandwidth extension based on GMM and clustering method," Proceedings of the Fifth International Conference on Communication Systems and Network Technologies (CSNT), pp.437–441, April 2015.

[119] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol.9, pp.1735–1780, 1997.

[120] D. Bansal, B. Raj, and P. Smaragdis, “Bandwidth expansion of narrowband speech using non-negative matrix factorization,” Proceedings of EUROSPEECH, 2005.

[121] T. Toda, A.W. Black, and K. Tokuda, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” IEEE Transactions on Audio, Speech, Language Process, vol.15, pp.2222–2235, Nov. 2007.

[122] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, “The Munich feature enhancement approach to the 2nd CHiME challenge using BLSTM recurrent neural networks,” Proceedings of the 2nd CHiME workshop on Machine Listening in Multisource Environments, pp.86–90, June 2013.

[123] 徳田恵一，小林隆夫，深田俊明，斎藤博徳，今井　聖，“メルケプストラムをパラメータとする音声のスペクトル推定,” 電子情報通信学会論文誌，vol.J74-A，pp.1240–1248，Aug. 1991.

[124] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, M. Petr, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, “The Kaldi speech recognition toolkit,” Proceedings of ASRU, pp.1–4, 2011.

[125] R. Haeb-Umbach and H. Ney, “Linear discriminant analysis for improved large vocabulary continuous speech recognition,” Proceedings of ICASSP, pp.13–16, 1992.

[126] R. Gopinath, “Maximum likelihood modeling with Gaussian distributions for classification,” Proceedings of ICASSP, pp.661–664, 1998.

[127] Y. Tachioka, T. Hanazawa, and T. Iwasaki, “Dereverberation method with reverberation time estimation using floored ratio of spectral subtraction,” Acoustical Science & Technology, vol.34, no.3, pp.212–215, 2013.

[128] 太刀岡勇気，花沢利行，成田知宏，石井　純，“音声と騒音の密度比推定を用いた音声区間検出法,” 電気学会論文誌 C，vol.133，pp.1549–1555，Aug. 2013.

[129] Y. Tachioka, T. Narita, and J. Ishii, “Estimation of speech recognition performance for clipped speech based on objective measures,” Acoustical Science & Technology, vol.35, no.6, pp.324–326, Nov. 2014.

[130] Y. Tachioka, T. Narita, T. Hanazawa, and J. Ishii, “Voice activity detection based on density ratio estimation and system combination,” Proceedings of APSIPAAPSIPA, pp.1–4 Nov. 2013.

[131] Y. Tachioka, T. Narita, I. Miura, T. Uramoto, N. Monta, S. Uenohara, K. Furuya, S. Watanabe, and J. Le Roux, “Coupled initialization of multi-channel non-negative matrix factorization based on spatial and spectral information,” Proceedings of INTERSPEECH, pp.2461–2465, Aug. 2017.

[132] J. Baker, L. Deng, J. Glass, S. Khudanpur, C. Lee, N. Morgan, and D. O'Shaughnessy, "Research developments and directions in speech recognition and understanding part 1," IEEE Signal Processing Magazine, vol.26, pp.75–80, May 2009.

[133] G. Saon and J.-T. Chien, "Large-vocabulary continuous speech recognition systems: A look at some recent advances," IEEE Signal Processing Magazine, vol.29, no.6, pp.18–33, 2012.

[134] M. Gales, "Semi-tied covariance matrices for hidden Markov models," IEEE Transactions on Speech and Audio Processing, vol.7, pp.272–281, March 1999.

[135] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," Proceedings of ICSLP, pp.1137–1140, 1996.

[136] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," Proceedings of ICASSP, pp.961–964, 2005.

[137] 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," IEEE Signal Processing Magazine, vol.28, pp.82–97, Nov. 2012.

[138] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," Technical Report, May 1997.

[139] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," Computer Speech and Language, vol.12, pp.75–98, April 1998.

[140] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," Computer Speech and Language, vol.9, pp.171–185, 1995.

[141] K. Shinoda and C. Lee, "A structural Bayes approach to speaker adaptation," IEEE Transactions on Speech and Audio Processing, vol.9, pp.276–287, 2001.

[142] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," IEEE Signal Processing Magazine, vol.25, pp.14–36, Sept. 2008.

[143] X. He and L. Deng, Discriminative Learning for Speech Recognition: Theory and Practice, Morgan & Claypool, 2008.

[144] L. Bahl, P. Brown, P. deSouza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," Proceedings of ICASSP, vol.11, pp.49–52, 1986.

[145] R. Schlüter, W. Macherey, B. Müller, and H. Ney, “Comparison of discriminative training criteria and optimization methods for speech recognition,” Speech Communication, vol.34, pp.287–310, May 2001.

[146] D. Povey and P. Woodland, “Minimum phone error and I-smoothing for improved discriminative training,” Proceedings of ICASSP, vol.I, pp.105–108, 2002.

[147] E. McDermott, T. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri, “Discriminative training for large-vocabulary speech recognition using minimum classification error,” IEEE Transactions on Audio, Speech, and Language Processing, vol.15, pp.203–223, Jan. 2007.

[148] L. Rabiner and B.-H. Juang, 音声認識の基礎，NTT アドバンステクノロジ株式会社，1995.

[149] 嵯峨山茂樹，“HMM(隠れマルコフモデル) とは何か,” 応用音響学資料，2008.

[150] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” Journal of the Royal Statistical Society SeriesB (Methodological), vol.39-1, pp.1–38, 1977.

[151] 赤穂昭太郎，“EM アルゴリズム：クラスタリングへの適用と最近の発展,” 日本知能情報ファジイ学会，vol.12-5，pp.2–10，2000.

[152] C.-H. Lee, F.K. Soong, and K.K. Paliwal, Automatic Speech and Speaker Recognition Advanced Topics, Kluwer Academic Publishers, 1996.

[153] N. Kanda, R. Takeda, and Y. Obuchi, “Elastic spectral distortion for lowresource speech recognition with deep neural networks,” Proceedings of ASRU, pp.309–314, Dec. 2013.

[154] F. Liu, R. Stern, X. Huang, and A. Acero, “Efficient cepstral normalization for robust speech recognition,” Proceedings of ARPA Workshop on Human Language Technology, pp.69–74, March 1993.

[155] O.M. Strand and A. Egeberg, “Cepstral mean and variance normalization in the model domain,” COST278 and ISCA Tutorial and Research workshop on Robustness Issues in Conversational Interaction, vol.4(38), Aug. 2004.

[156] S. Molau, M. Pitz, and H. Ney, “Histogram based normalization in the acoustic feature space,” Proceedings of ASRU, 2001.

[157] T.N. Sainath, B. Kingsbury, A.R. Mohamed, G.E. Dahl, G. Saon, H. Soltau, T. Beran, A.Y. Aravkin, and B. Ramabhadran, “Improvements to deep convolutional neural networks for LVCSR,” Proceedings of ASRU, pp.315–320, IEEE, Dec. 2013.

[158] T. Yoshioka, A. Ragni, and M. Gales, “Investigation of unsupervised adaptation of DNN acoustic models with filter bank input,” Proceedings of ICASSP, pp.13–16, May 2014.

[159] H. Kanagawa, Y. Tachioka, S. Watanabe, and J. Ishii, "Feature-space structural MAPLR with regression tree-based multiple transformation matrices for DNN," Proceedings of APSIPAAPSIPA, pp.86–92 Dec. 2015.

[160] T.N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," Proceedings of ICASSP, pp.8614–8618, May 2013.

[161] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," Proceedings of ICASSP, vol.4, pp.757–760, Honolulu, Hawaii, USA, 2007.

[162] D. Yu and M.L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," Proceedings of INTERSPEECH, pp.237–240, 2011.

[163] E. Eide and H. Gish, "A parametric approach to vocal tract length normalization," Proceedings of ICASSP, vol.1, pp.346–3483, 1996.

[164] M. Pitz, S. Molau, R. Schluter, R.S. Uter, and H. Ney, "Vocal tract normalization equals linear transformation in cepstral space," Proceedings of EUROSPEECH, pp.2653–2656, 2001.

[165] S. Umesh, A. Zolnay, and H. Ney, "Implementing frequency-warping and VTLN through linear transformation of conventional MFCC," Prodeedings of INTERSPEECH, pp.269–272, 2005.

[166] S. Panchapagesan and A. Alwan, "Frequency warping for VTLN and speaker adaptation by linear transformation of standard MFCC," Computer Speech and Language, vol.23, no.1, pp.42–64, Jan. 2009.

[167] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," Proceedings of ICASSP, vol.3, pp.1635–1638, Istanbul, June 2000.

[168] D. Ellis, R. Singh, and S. Sivadas, "Tandem acoustic modeling in large-vocabulary recognition," Proceedings of ICASSP, pp.517–520, 2001.

[169] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.ASP-34-1, pp.52–59, Feb. 1986.

[170] N. Kumar, Investigation of Silicon Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition, PhD Theis, Johns Hopkins University, 1997.

[171] N. Kumar and A.G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," Speech Communication, vol.26, pp.283–297, 1998.

[172] B. Zhang, S. Matsoukas, and R. Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction," Proceedings of INTERSPEECH, pp.1573–1576, 2006.

[173] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent feature transforms for speech recognition," Proceedings of ICASSP, vol.1, pp.313–316, 2006.

[174] B. Zhang and S. Matsoukas, "Minimum phoneme error based heteroscedastic linear discriminant analysis for speech recognition," Proceedings of ICASSP, pp.925–928, 2005.

[175] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," Proceedings of ICASSP, pp.4057–4060, 2008.

[176] J. Droppo and A. Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," Proceedings of INTERSPEECH, pp.989–992, 2005.

[177] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," Proceedings of ICASSP, pp.1129–1132, 2000.

[178] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[179] B. Chen, S.-H. Liu, and F.-H. Chu, "Training data selection for improving discriminative training of acoustic models," Pattern Recognition Letters, vol.30, pp.1228–1235, 2009.

[180] S. Furui, K. Maekawa, and H. Isahara, "A Japanese national project on spontaneous speech corpus and processing technology," Proceedings of ASR, pp.244–248, 2000.

[181] R. Hsiao, Generalized Discriminative Training for Speech Recognition, PhD thesis, Carnegie Mellon University, 2012.

[182] Y. Normandin and S.D. Morgera, "An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition," Proceedings of ICASSP, vol.1, pp.537–540, 1991.

[183] Y. Tachioka, S. Watanabe, J. Le Roux, and J. Hershey, "Discriminative methods for noise robust speech recognition: A CHiME challenge benchmark," Proceedings of the 2nd CHiME Workshop on Machine Listening in Multisource Environments, pp.19–24, June 2013.

[184] B. Kingsbury, T. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," Proceedings of INTERSPEECH, pp.485–488, Sept. 2012.

[185] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," Proceedings of INTERSPEECH, pp.2345–2349, Aug. 2013.

[186] D. Povey, "Improvements to fMPE for discriminative training of features," Proceedings of INTERSPEECH, pp.2977–2980, 2005.

[187] T. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," Proceedings of ICASSP, pp.6655–6659, May 2013.

[188] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," Proceedings of INTERSPEECH, pp.2365–2369, Aug. 2013.

[189] J. Bridle and L. Dodd, "An alphanet approach to optimising input transformations for continuous speech recognition," Proceedings of ICASSP, pp.277–280, 1991.

[190] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," Proceedings of ICASSP, pp.3761–3764, 2009.

[191] G. Wang and K. Sim, "Sequential classification criteria for NNs in automatic speech recognition," Proceedings of INTERSPEECH, pp.441–444, Aug. 2011.

[192] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech rrecognition," Proceedings of INTERSPEECH, pp.613–616, Sept. 2012.

[193] Y. Kubo, T. Hori, and A. Nakamura, "Large vocabulary continuous speech recognition based on WFST structured classifiers and deep bottleneck features," Proceedings of ICASSP, pp.7629–7633, May 2013.

[194] E. McDermott, Discriminative Training for Speech Recognition, Doctrorial dissertation, Waseda University, 1997.

[195] J. Fiscus, "A post-processing system to yield reduced error word rates: Recognizer output voting error reduction (ROVER)," Proceedings of ASRU, pp.347–354, Dec. 1997.

[196] H. Schwenk and J.-L. Gauvain, "Improved ROVER using language model information," Proceedings of ISCA ITRW Workshop on Automatic Speech Recognition, pp.47–52, Sept. 2000.

[197] G. Evermann and P. Woodland, "Posterior probability decoding, confidence estimation and system combination," Proceedings of NIST Speech Transcription Workshop, 2000.

[198] B. Hoffmeister, T. Klein, R. Schlüter, and H. Ney, "Frame based system combination and a comparison with weighted ROVER and CNC," Proceedings of ICSLP, pp.537–540, 2006.

[199] O. Siohan, B. Ramabhadran, and B. Kingsbury, "Constructing ensembles of ASR systems using randomized decision trees," Proceedings of ICASSP, pp.197–200, 2005.

[200] C. Breslin and M. Gales, "Generating complementary systems for speech recognition," Proceedings of INTERSPEECH, pp.525–528, 2006.

[201] H. Tang, M. Hasegawa-Johnson, and T.S. Huang, "Toward robust learning of the Gaussian mixture state emission densities for hidden Markov models," Proceedings of ICASSP, pp.5242–5245, 2010.

[202] G. Saon and H. Soltau, "Boosting systems for LVCSR," Proceedings of INTERSPEECH, pp.1341–1344, Sept. 2010.

[203] M.J.F. Gales, S. Watanabe, and E. Fosler-Lussier, "Structured discriminative models for speech recognition: An overview," IEEE Signal Processing Magazine, vol.29, pp.70–81, Nov. 2012.

[204] B. Roark, M. Saraçlar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," Proceedings of ACL, pp.47–54, 2004.

[205] F. Diehl and P. Woodland, "Complementary phone error training," Proceedings of INTERSPEECH, 2012.

[206] Y. Tachioka, S. Watanabe, J. Le Roux, and J. Hershey, "A generalized framework of discriminative training for system combination," Proceedings of ASRUIEEE, pp.43–48 2013.

[207] Y. Freund and R. Schapire, "A dicision-theoretic generalisation of online learning and an application to boosting," Journal of Computer and System Sciences, vol.55, pp.119–139, Aug. 1997.

[208] J. Friedman, T. Hestie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," Annals of Statistics, vol.28, pp.337–407, 2000.

[209] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," Proceedings of INTERSPEECH, pp.1045–1048, 2010.

[210] M. Sundermeyer, I. Oparin, J.-L. Gauvain, B. Freiberg, R. Schlüter, and H. Ney, "Comparison of feedforward and recurrent neural network language models," Proceedings of ICASSP, pp.8430–8434, May 2013.

[211] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, J. Černocký, and S. Khudanpur, "RNNLM–recurrent neural network language modeling toolkit," Proceedings of ASRU, pp.1–4, 2011.

[212] M. Sundermeyer, R. Schlüter, and H. Ney, "rwthlm - the RWTH Aachen university neural network language modeling toolkit," Proceedings of ICASSP, pp.2093–2097, May 2014.

[213] R. Iyer and M. Ostendorf, "Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models," Proceedings of ICSLP, vol.1, pp.236–239, 1996.

[214] Y. Shi, W.-Q. Zhang, M. Cai, and J. Liu, "Efficient one-pass decoding with NNLM for speech recognition," IEEE Signal Processing Letters, vol.21, pp.377–381, 2014.

[215] T. Hori, Y. Kubo, and A. Nakamura, "Real-time one-pass decoding with recurrent neural network language model for speech recognition," Proceedings of ICASSP, pp.6414–6418, May 2014.

[216] Z. Huang, G. Zweig, and B. Dumoulin, "Cache based recurrent neural network language model inference for first pass speech recognition," Proceedings of ICASSP, pp.6404–6407, May 2014.

[217] P. Brown, P. Desouza, R. Mercer, V. Pietra, and J. Lai, "Class-based n-gram models of natural language," Computational linguistics, vol.18, pp.467–479, 1992.

[218] Y. Tachioka, S. Watanabe, and J. Hershey, "Effectiveness of discriminative training and feature transformation for reverberated and noisy speech," Proceedings of ICASSP, pp.6935–6939 May 2013.

[219] Y. Tachioka, S. Watanabe, J. Le Roux, and J.R. Hershey, "Sequential maximum mutual information linear discriminant analysis for speech recognition," Proceedings of INTERSPEECH, pp.2415–2419, Sept. 2014.

[220] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," Proceedings of INTERSPEECH, pp.1209–1213, Sept. 2014.

[221] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," Proceedings of ICASSP, pp.4105–4109, May 2014.

[222] M. Federico, "Bayesian estimation methods of n-gram language model adaptation," Proceedings of ICSLP, pp.240–243, 1996.

[223] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," Computer Speech and Language, vol.10, pp.187–228, May 1996.

[224] R. Kneser, J. Peters, and D. Klakow, "Language model adaptation using dynamic marginals," Proceedings of EUROSPEECH, pp.1971–1974, 1997.

[225] H. Kuo, E. Fosler-Lussier, H. Jiang, and C. Lee, "Discriminative training of language models for speech recognition," Proceedings of ICASSP, vol.1, pp.325–328, 2002.

[226] T. Oba, T. Hori, A. Nakamura, and A. Ito, "Round-robin duel discriminative language models," IEEE Transactions on Audio, Speech, and Language Processing, vol.20, pp.1244–1255, May 2012.

[227] E. Dikici, M. Semarci, M. Saraçlar, and E. Alpaydin, "Classification and ranking approaches to discriminative language modeling for ASR," IEEE Transactions on Audio, Speech, and Language Processing, vol.21, pp.291–300, Feb. 2013.

[228] T. Oba, T. Hori, and A. Nakamura, "A study of efficient discriminative word sequences for reranking of recognition results based on n-gram counts," Proceedings of INTERSPEECH, pp.1753–1756, 2007.

[229] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," Proceedings of the ACL Conference on Empirical Methods in Natural Language ProcessingAssociation for Computational Linguistics, pp.1–8 2002.

[230] B. Juang and S. Katagiri, "Discriminative learning for minimum error classification [pattern recognition]," IEEE Transactions on Signal Processing, vol.40, pp.3043–3054, 1992.

[231] E. McDermott, S. Watanabe, and A. Nakamura, "Discriminative training based on an integrated view of MPE and MMI in margin and error space," Proceedings of ICASSP, pp.4894–4897, 2010.

[232] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," Speech Communication, vol.49, no.10, pp.827–835, Oct. 2007.

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

[234] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, "Is speech enhancement preprocessing still relevant when using deep neural networks for acoustic modeling?," Proceedings of INTERSPEECH, pp.2992–2996, Aug. 2013.

[235] A. Narayanan and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.22, pp.826–835, Feb. 2014.

[236] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," Speech Communication, vol.34, pp.267–285, 2001.

[237] H. Van hamme, "Robust speech recognition using missing feature theory in the cepstral or LDA domain," Proceedings of EUROSPEECH, 2003.

[238] D. Kolossa, A. Klimas, and R. Orglmeister, "Separation and robust recognition of noise, convolutive speech mixtures using time-frequency masking and missing data techniques," Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp.82–85, Oct. 2005.

[239] J. Arrowood and M. Clements, "Using observation uncertainty in HMM decoding," Proceedings of ICSLP, pp.1561–1564, 2002.

[240] H. Liao and M. Gales, "Joint uncertainty decoding for noise robust speech recognition," Proceedings of EUROSPEECH, pp.3129–3132, 2005.

[241] M. Delcroix, T. Nakatani, and S. Watanabe, "Static and dynamic variance compensation for recognition of reverberant speech with dereverberation preprocessing," IEEE Transactions on Audio, Speech, and Language Processing, pp.324–334, Feb. 2009.

[242] D. Kolossa, R.F. Astudillo, E. Hoffmann, and R. Orglmeister, "Independent component analysis and time-frequency masking for speech recognition in multi-talker conditions," EURASIP Journal on Audio, Speech, and Music Processing, p.ID 651420, 2010.

[243] R.F. Astudillo, Integration of Short-time Fourier Domain Speech Enhancement and Observation Uncertainty Techniques for Robust Automatic Speech Recognition, PhD Thesis, Universität Berlin, 2010.

[244] L. Lu, K. Chin, A. Ghoshal, and S. Renals, "Joint uncertainty decoding for noise robust subspace Gaussian mixture models," IEEE Transactions on Audio, Speech, and Language Processing, vol.21, pp.1791–1804, Sept. 2013.

[245] D.T. Tran, E. Vincent, and D. Jouvet, "Fusion of multiple uncertainty estimators and propagators for noise robust ASR," Proceedings of ICASSP, pp.5549–5553, May 2014.

[246] F. Nesta, M. Matassoni, and R.F. Astudillo, "A flexible spatial blind source extraction framework for robust speech recognition in noisy environments," Proceedings of the 2nd CHiME Workshop on Machine Listening in Multisource Environments, pp.33–38, June 2013.

[247] R. Astudillo and J. daSilva Neto, "Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition," Proceedings of INTERSPEECH, pp.461–464, 2011.

[248] D. Kolossa and R. Haeb-Umbach, Robust Speech Recognition of Uncertain or Missing Data – Theory and Applications, Springer Verlag, July 2011.

[249] W. Wright, "Bayesian approach to neural-network modeling with input uncertainty," IEEE Transactions on Neural Networks,, vol.10, no.6, pp.1261–1270, June 1999.

[250] S. Julier, J. Uhlmann, and H. Durrant-White, "A new method for non-linear transformation of means and covariances in filters and estimators," IEEE Transactions on Automatic Control, vol.45, pp.477–482, March 2000.

[251] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," Proceedings

of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3361–3368, 2011.

[252] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, “Extracting and composing robust features with denoising autoencoders,” Proceedings of the Twenty-fifth International Conference on Machine Learning, pp.1096–1103, 2008.

[253] A. Ozerov, E. Vincent, and F. Bimbot, “A general flexible framework for the handling of prior information in audio source separation,” IEEE Transactions on Audio, Speech and Language Processing, vol.20, pp.1118–1133, May 2012.

[254] 太刀岡勇気，渡部晋治，ルルージョナトン，ハーシージョン，“低ランク DNN 音響モデルの騒音下音声認識での評価と系列の識別学習,” 情報処理学会論文誌，vol.57，no.3，pp.1080–1088，March 2016.

[255] Y. Tachioka and J. Ishii, “Long short-term memory recurrent-neural-network-based bandwidth extension for automatic speech recognition,” Acoustical Science & Technology, vol.37, no.6, pp.319–321, Nov. 2016.

[256] Y. Tachioka and S. Watanabe, “Discriminative training of acoustic models for system combination,” Proceedings of INTERSPEECHISCA, pp.2355–2359 Aug. 2013.

[257] Y. Tachioka, S. Watanabe, J. Le Roux, and J. Hershey, “Sequential discriminative training for low-rank deep neural network,” Proceedings of the 2nd IEEE Global Conference on Signal and Information Processing (GlobalSIP)IEEE, pp.735–739 Dec. 2014.

[258] Y. Tachioka and S. Watanabe, “Discriminative method for recurrent neural network language models,” Proceedings of ICASSP, pp.5386–5390 April 2015.

[259] Y. Tachioka and S. Watanabe, “Uncertainty training and decoding methods of deep neural networks based on stochastic representation of enhanced features,” Proceedings of INTERSPEECH, pp.3541–3545, Sept. 2015.

[260] J. Barker, E. Vincent, N. Ma, C. Christensen, and P. Green, “The PASCAL CHiME speech separation and recognition challenge,” Computer Speech and Language, vol.27, no.3, pp.621–633, 2013.

[261] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, “The second ‘CHiME’ speech separation and recognition challenge: Datasets, tasks and baselines,” Proceedings of ICASSP, pp.126–130, May 2013.

[262] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj, “The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech,” Proceedings of WASPAA, pp.1–4, New Paltz, NY, USA, 2013.

[263] H. Christensen, J. Barker, N. Ma, and P. Green, "The CHiME corpus: a resource and a challenge for computational hearing in multisource environments," Proceedings of INTERSPEECH, pp.1918–1921, 2010.

[264] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," Computer Speech and Language, p.Elsevier, 2012.

[265] D. Johnson and D. Dudgeon, Array Signal Processing, Prentice-Hall, New Jersey, 1993.

[266] V. Goel and W. Byrne, "Minimum Bayes-risk automatic speech recognition," Computer Speech and Language, vol.14, pp.115–135, April 2000.

[267] W. Byrne, "Minimum Bayes risk estimation and decoding in large vocabulary continuous speech recognition," IEICE Transactions on Information and Systems, vol.E89-D, pp.900–907, 2006.

[268] H. Xu, D. Povey, L. Mangu, and J. Zhu, "An improved consensus-like method for minimum Bayes risk decoding and lattice combination," Proceedings of ICASSP, pp.4938–4941, 2010.

[269] H. Kuo, L. Mangu, E. Arisoy, and G. Saon, "Minimum Bayes risk discriminative language models for Arabic speech recognition," Proceedings of ASRU, pp.208–213, 2011.

[270] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," Journal of Acoustical Society of America, vol.87, pp.1738–1752, 1990.

[271] A. Deoras, D. Filimonov, M. Harper, and F. Jelinek, "Model combination for speech recognition using empirical Bayes risk minimization," Proceedings of Spoken Language Technology Workshop (SLT)IEEE, pp.235–240 2010.

[272] B. Ren, L. Wang, L. Lu, Y. Ueda, and A. Kai, "Combination of bottleneck feature extraction and dereverberation for distant-talking speech recognition," Multimedia Tools and Applications, vol.75, no.9, pp.5093–5108, 2016.

[273] A. Brutti and M. Matassoni, "On the use of early-to-late reverberation ratio for ASR in reverberant environments," Proceedings of ICASSP, pp.4638–4642 May 2014.

[274] A. Ogawa and A. Nakamura, "Joint estimation of confidence and error causes in speech recognition," Speech Communication, vol.54, pp.1014–1028, 2012.

[275] N. Dehak, Discriminative and generative approaches for long- and short-term speaker characteristics modeling: Application to speaker verification, Ph.D dissertation, cole de Technologies Suprieure, 2009.

[276] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Transactions on Audio, Speech, and Language Processing, vol.19, no.4, pp.788–798, May 2011.

[277] O. Siohan and M. Bacchiani, "iVector-based acoustic data selection," Proceedings of INTERSPEECH, Aug. 2013.

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

[279] S. Araki, H. Sawada, and S. Makino, "Blind speech separation in a meeting situation with maximum SNR beamformers," Prodeedings of ICASSP, vol.1, pp.41–45, 2007.

[280] Y. Tachioka, T. Narita, S. Watanabe, and F. Weninger, "Dual system combination approach for various reverberant environments," Proceedings of REVERB challenge workshop, pp.1–8, May 2014.

[281] H. Hermansky, L. Burget, J. Cohen, E. Dupoux, N. Feldman, J. Godfrey, S. Khudanpur, M. Maciejewski, S. Mallidi, A. Menon, T. Ogawa, V. Peddinti, R. Rose, R. Stern, M. Wiesner, and K. Vesely, "Towards machines that know when they do not know: Summary of work done at 2014 Frederick Jelinek memorial workshop," Proceedings of ICASSP, pp.5009–5013 April 2015.

[282] B. Loesch and B. Yang, "Adaptive segmentation and separation of determined convolutive mixtures under dynamic conditions," Proceedings of the 9th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), pp.41–48, 2010.

[283] X. Mestre and M.A. Lagunas, "On diagonal loading for minimum variance beamformers," Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)IEEE, pp.459–462 2003.

[284] R. Monzingo and T. Miller, Introduction to Adaptive Arrays, Wiley and Sons, New York, 1980.

[285] 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, vol.46, pp.535–557, 2016.

[286] J. Eggert and E. Komer, "Sparse coding and NMF," Proceedings of the IEEE International Joint Conference on Neural Networks, vol.4, pp.2529–2533, IEEE, July 2004.

[287] T. Ochiai, S. Matsuda, H. Watanabe, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training for deep neural networks embedding linear transformation networks," Proceedings of ICASSP, pp.4605–4609, IEEE, April 2015.

[288] T. Hori, C. Hori, S. Watanabe, and J. Hershey, "Minimum word error training of long short-term memory recurrent neural network language models for speech recognition," Proceedings of ICASSP, pp.5990–5994, IEEE, March 2016.

[289] G. Saon, S. Dharanipragada, and D. Povey, "Feature space Gaussianization," Proceedings of ICASSP, vol.I, pp.329–332, 2004.

[290] K. Palomäki and H. Kallasjoki, "Reverberation robust speech recognition by matching distributions of spectrally and temporally decorrelated features," Proceedings of REVERB Workshop, 2014.

[291] D. Povey and K. Yao, "A basis representation of constrained MLLR transforms for robust adaptation," Computer Speech and Language, vol.26, pp.35–51, 2012.

[292] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," Proceedings of ICASSP, pp.4273–4276, 2012.

[293] K. Audhkhasi, A. Zavou, P. Georgiou, and S. Narayanan, "Theoretical analysis of diversity in an ensemble of automatic speech recognition systems," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.22, no.3, pp.711–726, March 2014.

[294] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model –a structured model for speech recognition," Computer Speech and Language, vol.25, no.2, pp.404–439, April 2011.

[295] B.E. Kingsbury and N. Morgan, "Recognizing reverberant speech with RASTA-PLP," Proceedings of ICASSP, vol.2, pp.21–24, April 1997.

[296] A.H. Sayed, Adaptive Filters, John Wiley & Sons, New Jersey, 2008.

[297] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAMO: a British English speech corpus for large vocabulary continuous speech recognition," Proceedings of ICASSP, pp.81–84, 1995.

[298] J. Snoek, H. Larochelle, and R. Adams, "Practical bayesian optimization of machine learning algorithms," Proceedings of Neural Information Processing Systems, 2012.

[299] G.E. Dahl, T.N. Sainath, and G.E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," Proceedings of ICASSP, pp.8609–8613, IEEE, May 2013.

[300] S. Watanabe and J. Le Roux, "Black box optimization for automatic speech recognition," Proceedings of ICASSP, pp.3280–3284, May 2014.

[301] F. Weninger, J.R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," Proceedings of GlobalSIP, pp.740–744, 2014.

[302] L. Cristoforetti, M. Ravanelli, M. Omologo, A. Sosi, A. Abad, M. Hagmueller, and P. Maragos, "The DIRHA simulated corpus," Proceedings of LREC, pp.2629–2634, May 2014.

[303] M. Fujimoto and K. Ishizuka, “Noise robust voice activity detection based on switching Kalman filter,” IEICE Transactions on Information and Systems, vol.E91-D, pp.467–477, March 2008.

[304] H. Do, H. Silverman, and Y. Yu, “A real-time SRP-PHAT source location implementation using stochastic region contraction(src) on a large-aperture microphone array,” Proceedings of ICASSP, vol.1, pp.121–124, April 2007.

[305] Y. Tachioka, T. Narita, and S. Watanabe, “Effectiveness of dereverberation, feature transformation, discriminative training methods, and system combination approach for various reverberant environments,” EURASIP Journal on Advances in Signal Processing, June 2015.

[306] Y. Tachioka, S. Watanabe, J. Le Roux, and J. Hershey, “Prior-based binary masking and discriminative methods for reverberant and noisy speech recognition using distant stereo microphones,” Journal of Information Processing, vol.25, no.6, pp.407–416, June 2017.

[307] Y. Tachioka, T. Narita, S. Watanabe, and J. Le Roux, “Ensemble integration of calibrated speaker localization and statistical speech detection,” Proceedings of the 4th workshop on Hands-free Speech Communication and Microphone Array (HSCMA)IEEE, pp.1–5 May 2014.

[308] Y. Tachioka and T. Narita, “Optimal automatic speech recognition system selection for noisy environments,” Proceedings of APSIPA, pp.1–8, Dec. 2016.

[309] 片木孝至，“アンテナ技術者の一人として，” 通信ソサイエティマガジン，vol.10，pp.4–11，2009.

[310] Y. Tachioka, Y. Yasuda, and T. Sakuma, “Application of the constrained interpolation profile method to room acoustic problems – Examination of boundary modeling and spatial/time discretization,” Acoustical Science & Technology, vol.33, no.1, pp.21–32, Jan. 2012.

# Publication list

The publications related to this thesis are as follows.

## Journal papers

(1), (2), (4), (5), and (8) are full papers and the others are short papers.

(1) Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey: Prior-based binary masking and discriminative methods for reverberant and noisy speech recognition using distant stereo microphones, Journal of Information Processing vol.25 no.6, pp.407-416, 2017. 6.

(2) Y. Tachioka and T. Narita: Template-based method for compensation of time difference of arrival in passive sound source localization under reverberant and noisy environments, Journal of Signal Processing vol.21 no.2 pp.73-79, 2017. 3.

(3) Y. Tachioka and J. Ishii: Long short-term memory recurrent-neural-network-based bandwidth extension for automatic speech recognition, Acoustical Science & Technology vol.37 no.6, pp.319-321, 2016. 11.

(4) 太刀岡 勇気, 渡部 晋治, ルルー ジョナトン, ハーシー ジョン: 低ランク DNN 音響モデルの騒音下音声認識での評価と系列の識別学習, 情報処理学会論文誌 vol.57 no.3, pp.1080-1088, 2016. 3.

(5) Y. Tachioka, T. Narita, and S. Watanabe: Effectiveness of dereverberation, feature transformation, discriminative training methods, and system combination approach for various reverberant environments, EURASIP Journal on Advances in Signal Processing, 2015:52 doi:10.1186/s13634-015-0241-y, 2015. 6.

(6) Y. Tachioka, T. Narita, and J. Ishii: Semi-blind source separation using binary masking and independent vector analysis, IEEJ Transactions on Electrical and Electronic Engineering vol.10 no.1, pp.114-115, 2015. 1.

(7) Y. Tachioka, T. Narita, and J. Ishii: Estimation of speech recognition performance for clipped speech based on objective measures, Acoustical Science & Technology vol.35 no.6, pp.324-326, 2014. 11.

(8) 太刀岡 勇気, 花沢 利行, 成田 知宏, 石井 純: 音声と騒音の密度比推定を用いた音声区間検出法, 電気学会論文誌 C (電子・情報・システム部門誌) vol.133 no.8, pp.1549-1555, 2013. 8.

(9) Y. Tachioka, T. Hanazawa, and T. Iwasaki: Dereverberation method with reverberation time estimation using floored ratio of spectral subtraction, Acoustical Science & Technology vol.34 no.3, pp.212-215, 2013. 5.

(10) Y. Tachioka, T. Narita, and T. Iwasaki: Direction of arrival estimation by cross-power spectrum phase analysis using prior distributions and voice activity detection information, Acoustical Science & Technology vol.33 no.1, pp.68-71, 2012. 1.

## Peer-reviewed conference papers

(1) Y. Tachioka, T. Narita, I. Miura, T. Uramoto, N. Monta, S. Uenohara, K. Furuya, S. Watanabe, and J. Le Roux: Coupled initialization of multi-channel non-negative matrix factorization based on spatial and spectral information, The 18th Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, 2017. 8.

(2) Y. Tachioka and T. Narita: Optimal automatic speech recognition system selection for noisy environments, Asia-Pacific Signal and Information Processing Association (APSIPA) Annual Summit and Conference, Jeju, pp.1-8, 2016. 12.

(3) Y. Tachioka, S. Watanabe, and T. Hori: The MELCO/MERL system combination approach for the fourth CHiME challenge, The Fourth CHiME Challenge Workshop, San Francisco, pp.1-3, 2016. 9.

(4) Y. Tachioka and S. Watanabe: Uncertainty training and decoding methods of deep neural networks based on stochastic representation of enhanced features, The 16th Annual Conference of the International Speech Communication Association (INTERSPEECH), Dresden, pp.3541-3545, 2015. 9.

(5) Y. Tachioka and S. Watanabe: Discriminative method for recurrent neural network language models, The 40th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brisbane, pp.5386-5390, 2015. 4.

(6) Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey: Sequential discriminative training for low-rank deep neural network, The 2nd IEEE Global Conference on Signal and Information Processing (GlobalSIP), Atlanta, pp.735-739, 2014. 12.

(7) Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey: Sequential maximum mutual information linear discriminant analysis for speech recognition, The 15th Annual Conference of the International Speech Communication Association (INTERSPEECH), Singapore, pp.2415-2419, 2014. 9.

(8) Y. Tachioka, T. Narita, S. Watanabe, and J. Le Roux: Ensemble integration of calibrated speaker localization and statistical speech detection, The 4th workshop on Hands-free Speech Communication and Microphone Array (HSCMA), Nancy, ID 1569901761 pp.1-5, 2014. 5.

(9) Y. Tachioka, T. Narita, S. Watanabe, and F. Weninger: Dual system combination approach for various reverberant environments, REVERB Challenge Workshop, Florence, ID 1569886337 pp.1-8, 2014. 5.

(10) Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey: A generalized framework of discriminative training for system combination, Automatic Speech Recognition and Understanding Workshop (ASRU), Olomouc, pp.43-48, 2013. 12.

(11) Y. Tachioka, T. Narita, T. Hanazawa, and J. Ishii: Voice activity detection based on density ratio estimation and system combination, Asia-Pacific Signal and Information Processing Association (APSIPA) Annual Summit and Conference, Kaohsiung, pp.1-4, 2013. 11.

(12) Y. Tachioka and S. Watanabe: Discriminative training of acoustic models for system combination, The 14th Annual Conference of the International Speech Communication Association (INTERSPEECH), Lyon, pp.2355-2359, 2013. 8.

(13) Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey: Discriminative methods for noise robust speech recognition: A CHiME Challenge Benchmark, The 2nd International Workshop on Machine Listening in Multisource Environments, Vancouver, pp.19-24, 2013. 6.

(14) Y. Tachioka, S. Watanabe, and J. R. Hershey: Effectiveness of discriminative training and feature transformation for reverberated and noisy speech, The 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, pp.6935-6939, 2013. 5.

## Non-reviewed conference papers

(1) 太刀岡 勇気, 渡部 晋治，ルルー ジョナトン，ハーシー ジョン: DNN の低ランク近似と識別学習の組み合わせ法, 電子情報通信学会総合大会, 名城大学 (天白), p.143, 2017. 3.

(2) 太刀岡 勇気，渡部 晋治: 強調音声の効率的サンプリングによる DNN の不確定性学習とデコード法, 日本音響学会研究発表会講演論文集 (秋季), 富山大, pp.27-30, 2016. 9.

(3) 太刀岡 勇気, 石井 純: 音声認識のための LSTM-RNN を用いた帯域拡張, 電子情報通信学会総合大会, 九大 (伊都キャンパス), p.122, 2016. 3.

(4) 太刀岡 勇気, 渡部 晋治，ルルー ジョナトン，ハーシー ジョン: 音声認識のための線形判別分析の系列相互情報量最大化識別学習, 日本音響学会研究発表会講演論文集 (春季), 桐蔭横浜大, pp.37-40, 2016. 3.

(5) 太刀岡 勇気，渡部 晋治: リカレントニューラルネットワーク言語モデルの識別学習, 日本音響学会研究発表会講演論文集 (秋季), 会津大, pp.13-16, 2015. 9.

(6) 太刀岡 勇気, 成田 知宏, 渡部 晋治, Jonathan Le Roux: 家庭環境における補正音源定位と統計的音声区間検出の統合 － DIRHA コーパスの利用－, 日本音響学会研究発表会講演論文集 (春季), 中央大 (後楽園), pp.591-594, 2015. 3.

(7) 太刀岡 勇気, 成田 知宏, 石井 純: 独立ベクトル分析のリアルタイム化, 電子情報通信学会総合大会, 立命館大 (びわこ・くさつキャンパス), p.94, 2015. 3.

(8) 太刀岡 勇気, 成田 知宏, 渡部 晋治, Felix Weninger: 残響環境下音声認識に対する残響除去とシステム統合手法の有効性 REVERB チャレンジ, 日本音響学会研究発表会講演論文集 (秋季), 北海学園大, pp.19-22, 2014. 9.

(9) 太刀岡 勇気, 渡部 晋治, Jonathan Le Roux, John R. Hershey: 音声認識システムの統合を目的とした識別学習の枠組み, 日本音響学会研究発表会講演論文集 (秋季), 北海学園大, pp.3-6, 2014. 9.

(10) 太刀岡 勇気, 成田 知宏, 石井 純: 2 値マスクと独立ベクトル分析を併用したセミブラインド音源分離, 電子情報通信学会総合大会, 新潟大, p.61, 2014. 3.

(11) 太刀岡 勇気, 渡部 晋治, Jonathan Le Roux, John R. Hershey: システム統合のための音響モデルの相互情報量最大化識別学習, 日本音響学会研究発表会講演論文集 (春季), 日大 (理工), pp.35-38, 2014. 3.

(12) 太刀岡 勇気, 渡部 晋治, Jonathan Le Roux, John R. Hershey: 騒音環境に対する識別的アプローチの有効性: 第 2 回 CHiME チャレンジ, 日本音響学会研究発表会講演論文集 (秋季), 豊橋技科大, pp.1-4, 2013. 9.

(13) 太刀岡 勇気, 成田 知宏, 石井 純: クリップした音声の音声認識, 日本音響学会研究発表会講演論文集 (春季), 東京工科大 (八王子), pp.13-14, 2013. 3.

(14) 太刀岡 勇気, 成田 知宏, 石井 純: 音源距離推定方式の比較検討とコスト関数の一般化, 日本音響学会研究発表会講演論文集 (秋季), 信州大, pp.90-93, 2012. 9.

(15) 太刀岡 勇気, 花沢 利行, 成田 知宏, 石井 純: 音声と騒音の密度比推定を用いた音声区間検出法, 日本音響学会研究発表会講演論文集 (春季), 神奈川大 (横浜), pp.9-12, 2012. 3.

(16) 太刀岡 勇気, 成田 知宏, 岩崎 知弘: 事前分布を用いた CSP 法による到来音方向推定, 日本音響学会研究発表会講演論文集 (春季), 早大, pp.661-664, 2011. 3.

(17) 太刀岡 勇気, 花沢 利行, 岩崎 知弘: 拡散音場理論に基づく残響環境下音声認識の検討 － 騒音環境下での評価 －, 日本音響学会研究発表会講演論文集 (春季), 電通大, pp.17-20, 2010. 3.

(18) 太刀岡 勇気, 花沢 利行, 岩崎 知弘: 拡散音場理論に基づく残響環境下音声認識の検討, 日本音響学会研究発表会講演論文集 (秋季), 日大 (郡山）, pp.35-38, 2009. 9.

## Non-reviewed workshop papers

(1) 太刀岡 勇気，渡部 晋治: 音声認識のための再帰的ニューラルネット言語モデルの識別学習, 電子情報通信学会技術研究報告 116(165), SP2016-26, 天童温泉 (滝の湯), pp.33-38, 2016. 7.

(2) 太刀岡 勇気，成田 知宏，渡部 晋治，ルルー ジョナトン: 音源定位と音声区間検出の有機的統合 –家庭環境を対象に–, SIP シンポジウム資料, スパリゾートハワイアンズ, pp.224-229, 2015. 11.

(3) 太刀岡 勇気，渡部 晋治，ルルー ジョナトン，ハーシー ジョン: 低ランク DNN のための識別学習, 電子情報通信学会技術研究報告, SP2014-39, かたくら諏訪湖ホテル, pp.19-24, 2015. 7.

(4) 太刀岡 勇気, 成田 知宏, 渡部 晋治: 残響除去手法とシステム統合手法の種々の残響環境に対する有効性: REVERB チャレンジ, 情報処理学会研究報告, SLP-105(6), 伊勢志摩 合歓の郷ホテル＆リゾート, pp.1-6, 2015. 2.

(5) 太刀岡 勇気, 渡部 晋治, J. Le Roux, J. R. Hershey: システム統合を目的とした識別学習の一般的枠組み, 電子情報通信学会技術研究報告, SP2014-65, 花巻温泉 (ホテル花巻), pp.13-18, 2014. 7.

(6) 太刀岡 勇気, 渡部 晋治, J. Le Roux, J. R. Hershey: システム統合のための音響モデルの識別学習, 電子情報通信学会技術研究報告, SP2014-15, 日本大 (文理), pp.147-152, 2014. 5.

(7) 太刀岡 勇気, 渡部 晋治, J. Le Roux, J. R. Hershey: 騒音環境に対する識別的アプローチの有効性: 第 2 回 CHiME チャレンジ, 電子情報通信学会技術研究報告, SP2013-55, 遠刈田温泉 (壮鳳), pp.13-18, 2013. 7.

(8) 太刀岡 勇気, 花沢 利行, 成田 知宏, 石井 純: 音声と騒音の密度比推定を用いた音声区間検出法, 電子情報通信学会技術研究報告, SP2012-54, 天童温泉 (滝の湯), pp.23-28, 2012. 7.

(9) 太刀岡 勇気, 花沢 利行, 岩崎 知弘: 拡散音場理論に基づく残響環境下音声認識 [オーガナイズドセッション：スピーチエンハンスメント], 電子情報通信学会技術研究報告, SP2010-4, 甲南大 (神戸・平生セミナーハウス), pp.19-24, 2010. 5.