SPECTRAL MAPPING FOR VOICE CONVERSION

ZHIZHENG WU

A thesis submitted to the School of Computer Engineering in partial fulfilment of the requirements for the degree of Doctor of Philosophy

March 2015
Abstract

Voice conversion is the process to modify a speech signal of one speaker (source) to sound like an intended speaker (target) without changing the language content. This technology has several applications, such as personalized speech synthesis and speech to singing, and a possible malicious application to fool a voice biometric system, also called a speaker verification system. This thesis focuses on techniques to improve voice conversion performance and the use of voice conversion technology to attack speaker verification systems.

The study is important as the robustness of the conversion function will affect the performance of voice conversion directly. To this end, the thesis first proposes a method to improve the voice conversion function by benefiting from nonparallel data of background speakers. The strategy is to decompose a speech spectral vector into a phonetic component and a speaker-specific component, which are modeled by a factor analysis model. The nonparallel data are used to estimate the phonetic component and the factor loadings. The speaker-specific component can then be represented by a low-dimensional set of variables via factor loadings, and hence allow for more robust modeling under sparse training data condition. The experimental results show that the proposed method outperforms the conventional Gaussian mixture model (GMM) based method considerably when there are limited parallel training data.

The second contribution of the thesis focuses on implementing the conversion function by directly modelling the high-dimensional spectral features using exemplars found in the training data. In this approach, each speech segment is reconstructed as a weighted linear combination of a set of basis exemplars with residual compensation. An exemplar is defined as a speech segment spanning multiple frames extracted from training data. The value of the linear combination weights is constrained to be nonnegative, and most of them are restricted to have a value close to zero. Experiments are
conducted to compare the proposed method with a large set of baseline approaches. It is observed that the proposed method can achieve similar performance to the state-of-the-art GMM based and dynamic kernel partial least square based voice conversion methods. The experiments also confirm its flexibility when the amount of training data is varied.

The third contribution focuses on the use of voice conversion technology to attack current state-of-the-art speaker verification system with the purpose to identify the weak links of the verification algorithms. Recently, speaker verification technology has been advanced significantly and has led to mass market adoption, such as in smartphones for user authentication. A major concern when deploying speaker verification technology is whether a system is still robust against spoofing attacks. Unfortunately, voice conversion has become one of the most easily accessible techniques to carry out spoofing attacks, and hence presents a threat to speaker verification systems. To address the above concern, this thesis examines the vulnerabilities of nine current state-of-the-art speaker verification systems in the face of voice conversion spoofing attacks.
Acknowledgements

I would like to express my sincere thanks to my supervisors, Dr. Chng Eng Siong and Dr. Li Haizhou for the time and effort they spent brainstorming ideas and offering me guidance. I appreciate that they always encourage me to pursue my interests and explore new research areas.

I also want to thank Dr. Tomi Kinnunen, Dr. Tuomas Virtanen, Dr. Kong-Aik Lee, and Dr. Anthony Larcher for their help and discussions. Tomi helps me a lot to improve my writing skills and to organize my ideas. Tuomas teaches how to apply sparse coding techniques to voice conversion and how to do analysis and set up experiments. Without Kong-Aik and Anthony’s support, I am not able to conduct the spoofing attack research.

My thanks also go to my colleagues in Speech and Language Technology Program at NTU and my friends in C2I lab. My life is much more colorful and comfortable with their friendship.

I also want to thank my colleagues in human language technique dependent of I2R, including Dr. Ma Bin, Dr. Dong Minghui, Dr. Wang Lei, Dr. Lee Siu Wa and Dr. Gao Sheng for their help.

Finally, I want to thank my family in China and my wife for their encouragements and constant love.
Contents

Abstract ................................................................. ii
Acknowledgements ..................................................... iv
List of Publications .................................................. viii
List of Figures ........................................................ xi
List of Tables .......................................................... xiv

1 Introduction .......................................................... 1
  1.1 Contributions .................................................... 2
  1.2 Thesis outline ................................................... 4

2 Background .......................................................... 5
  2.1 Speech production and speaker identity ....................... 5
  2.2 Overview of voice conversion ................................ 10
    2.2.1 Speech analysis and reconstruction ..................... 12
    2.2.2 Feature extraction ........................................ 14
    2.2.3 Frame alignment .......................................... 15
    2.2.4 Prosody conversion ....................................... 17
  2.3 Spectral mapping techniques ................................ 18
    2.3.1 Vector quantization (VQ) ................................ 19
    2.3.2 Gaussian mixture model based methods ................ 20
    2.3.3 Partial least squares regression based methods ....... 25
    2.3.4 Neural network based methods ........................... 27
    2.3.5 Frequency warping based methods ....................... 29
    2.3.6 Unit selection based methods ........................... 31
  2.4 Applications ..................................................... 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1 Text-to-speech synthesis</td>
<td>33</td>
</tr>
<tr>
<td>2.4.2 Spoofing attack</td>
<td>34</td>
</tr>
<tr>
<td>2.5 Summary</td>
<td>35</td>
</tr>
<tr>
<td>3 Mixture of Factor Analysers Using Priors from Nonparallel Speech</td>
<td>36</td>
</tr>
<tr>
<td>3.1 Motivation</td>
<td>36</td>
</tr>
<tr>
<td>3.2 Mixture of factor analyzers</td>
<td>38</td>
</tr>
<tr>
<td>3.2.1 Speaker-independent phonetic vectors estimation</td>
<td>39</td>
</tr>
<tr>
<td>3.2.2 Speaker-independent factor loadings estimation</td>
<td>41</td>
</tr>
<tr>
<td>3.2.3 Voice conversion using mixture of factor analyzers</td>
<td>41</td>
</tr>
<tr>
<td>3.3 Experiments</td>
<td>43</td>
</tr>
<tr>
<td>3.3.1 Objective evaluation</td>
<td>44</td>
</tr>
<tr>
<td>3.3.2 Subjective evaluation</td>
<td>46</td>
</tr>
<tr>
<td>3.4 Conclusion</td>
<td>48</td>
</tr>
<tr>
<td>4 Exemplar-based sparse representation with residual compensation for</td>
<td>49</td>
</tr>
<tr>
<td>voice conversion</td>
<td></td>
</tr>
<tr>
<td>4.1 Motivation</td>
<td>50</td>
</tr>
<tr>
<td>4.2 Exemplar-based sparse representation with residual compensation</td>
<td>52</td>
</tr>
<tr>
<td>4.2.1 Basic exemplar-based sparse representation</td>
<td>52</td>
</tr>
<tr>
<td>4.2.2 Spectrum compression</td>
<td>54</td>
</tr>
<tr>
<td>4.2.3 Contextual information</td>
<td>55</td>
</tr>
<tr>
<td>4.2.4 Using low-resolution features for faster computation</td>
<td>56</td>
</tr>
<tr>
<td>4.2.5 Compensating model residual</td>
<td>57</td>
</tr>
<tr>
<td>4.2.6 Dictionary Construction</td>
<td>59</td>
</tr>
<tr>
<td>4.3 Experiments</td>
<td>60</td>
</tr>
<tr>
<td>4.3.1 Reference methods and setups</td>
<td>60</td>
</tr>
<tr>
<td>4.3.2 Objective evaluations</td>
<td>63</td>
</tr>
<tr>
<td>4.3.3 Subjective evaluations</td>
<td>71</td>
</tr>
<tr>
<td>4.4 Discussion</td>
<td>79</td>
</tr>
<tr>
<td>4.4.1 Relationship with vector quantization and frame selection methods</td>
<td>79</td>
</tr>
<tr>
<td>4.4.2 Relationship with JD-GMM methods</td>
<td>80</td>
</tr>
<tr>
<td>4.4.3 Relationship with PLS</td>
<td>81</td>
</tr>
</tbody>
</table>
4.4.4 Computational complexity and memory footprint ................. 82
4.5 Conclusions ........................................................................ 83

5 Vulnerability Evaluation of Speaker Verification Under Voice Conversion Spoofing Attacks ............................................................................................................. 85

5.1 Motivation ................................................................................. 86
5.2 Related work on speaker verification vulnerabilities ......................... 88
5.3 Voice conversion system ................................................................. 90
5.4 Speaker Verification Systems ....................................................... 92
5.4.1 Text-independent speaker verification systems ............................ 92
5.4.2 Text-dependent speaker verification systems .............................. 93
5.4.3 Points of speaker verification vulnerabilities ................................ 95
5.5 Experimental protocol ................................................................. 96
5.5.1 Performance evaluation measures ............................................ 96
5.5.2 Baseline and spoofing datasets ................................................. 97
5.6 Experimental results and discussion ............................................ 100
5.6.1 Parameterizations ................................................................. 101
5.6.2 Case study 1: Attacking text-independent systems ..................... 102
5.6.3 Case study 2: Attacking text-dependent systems ....................... 103
5.6.4 Case study 3: Voice conversion performance vs spoofing effect .... 107
5.7 Conclusions ............................................................................. 108

6 Conclusions and Future Work ...................................................... 112

6.1 Contributions ........................................................................... 112
6.1.1 Mixture of factor analyzers using nonparallel data .................. 113
6.1.2 Exemplar-based sparse representation with residual compensation 114
6.1.3 Voice conversion spoofing attacks to speaker verification systems 115
6.2 Future Directions ..................................................................... 116

References .................................................................................. 118
List of Publications

Journal


Conference


List of Figures

2.1 Illustration of the human organs associated with the human speech production process [1]. .................................................. 6
2.2 A voiced sound example of the phoneme /i:/, the signal of which is periodic. The left figure is the time-domain representation of the speech signal, also known as waveform representation. The right figure is the frequency-domain representation of the same speech signal, known as spectrum and corresponding spectral envelope. ............................. 7
2.3 An unvoiced sound example of the phoneme /s/. The left figure is the waveform which looks like random noise, and the right figure is the corresponding spectrum representation. ................................. 8
2.4 Comparison of male (left) and female (right) speech signals to illustrate the difference of speech representations across speakers. (a) Time-domain waveform representation; (b) Frequency-domain spectrogram representation; (c) The first four formant tracks, F1-F4; (d) Fundamental frequency trajectories; (e) Intensity trajectories. These layers of information are aligned with corresponding phonemes. Praat [2] is used to create the figure. .............................................................. 9
2.5 A typical framework of a stand-alone voice conversion system, which consists of offline training and runtime conversion phases. ......................... 11
2.6 Illustration of joint vector quantization. The square points are entries for source feature codebook while circle points stand for entries for target feature codebook. Note that the square points and circle points always appear in pairs to stand for joint vectors. ..................................... 19
2.7 An example of the covariance matrix from Gaussian mixture model. . 24
2.8 An example of a cross diagonal matrix, in which only diagonal and
cross diagonal elements are non-zeros. ........................................... 25
2.9 Illustration of the dynamic kernel partial least squares regression for
voice conversion. MCC=Mel-cepstral coefficient. ................................. 27
2.10 Illustration of a frequency warping function to warp the frequency axes. 30
2.11 Illustration of searching for the optimal exemplar sequence. In the
figure, dashed line (connecting $X(t)$ and $Y'(t)$) represents target cost and
solid line (connecting $Y'_k(t)$ and $Y'_{(t+1)}$) represents concatenation cost. 33
2.12 Illustration of one way to integrate voice conversion with a unit-selection
based text-to-speech synthesis system. ............................................. 34
2.13 Diagram of a speaker verification system. ...................................... 34
3.1 Proposed spectral conversion system ............................................ 40
3.2 Average mel-cepstral distortion as a function of the number of factors. 45
3.3 Average MCD in terms of number of training utterances ................. 46
3.4 AB preference test results for speaker similarity with 95% confidence
intervals. ......................................................................................... 47
3.5 AB preference test results for speech quality with 95% confidence intervals. 48
4.1 Illustration of the smoothing effect of low-resolution features. The
dashed line is reconstructed from 24-order MCCs, which are computed
from the solid line. ................................................................. 51
4.2 Illustration of spectral shapes with corresponding compression factors ($\rho$). 55
4.3 Illustration of the exemplar-based sparse representation with residual
compensation framework, which consists of four processes: (1) estimating
the activation matrix $\mathbf{H}$ using low-resolution features; (2) calculating
the source residuals $\mathbf{R}^{(X)}$ following Eq. (4.11); (3) mapping source residuals to target residuals following Eq. (4.13); and (4) generating
converted spectrograms $\tilde{\mathbf{Y}}^{(DFT)}$ following Eq. (4.14). In the diagram,
$U$ and $F$ are the feature dimensionalities of low-resolution and high-
resolution features, respectively, $3 \times U$ means an exemplar spanning
three frames, $M$ is the number of frames in a source utterance, and $N$ is
the number of exemplars in a dictionary. As discussed in Section 4.2.6, $U$ is set to 50, and $F$ is set to 513. ............................... 57
4.4 Spectral distortion as a function of the number of exemplars $N$ in a dictionary. .............................................. 64
4.5 Spectral distortion as a function of spectral compression factors ($\rho$). .............................................. 66
4.6 Computing time to generate one second of speech as a function of the window size ($L$) of an exemplar. .............................................. 67
4.7 Spectral distortion as a function of the window size ($L$) of an exemplar. .............................................. 67
4.8 Illustration of the activation weights associated to each exemplar to generate a target spectrum. .............................................. 68
4.9 Illustration of top 100 activation weights out of the 3000 exemplars, which are averaged over an utterance. .............................................. 69
4.10 Spectral distortions on the evaluation set as a function of the number of exemplars/frames $N$ as training. .............................................. 71
4.11 Preference test results of speech quality with 95% confidence intervals for NMF-DFT and NMF-DFT-SC methods. .............................................. 72
4.12 Preference test results of speech quality with 95% confidence intervals for NMF-DFT-SC and NMF-MEL-SC methods. .............................................. 73
4.13 Preference test results of speech quality with 95% confidence intervals for NMF-MEL-SC with and without multiple-frame exemplars. ............ 73
4.14 Preference test results of speech quality with 95% confidence intervals for NMF-MEL-SC and NMF-MEL-SC-RC methods. .............................................. 74
4.15 Preference scores of speech quality with 95% confidence intervals for PLS-based methods with $N \approx 5500$ training frames and our proposed NMF-MEL-SC-RC method with $N = 3000$ exemplars in the dictionary. From top to bottom, (a), DKPLS vs. NMF-MEL-SC-RC; (b), DKPLS-DFT vs. NMF-MEL-SC-RC; and (c), DPLS-DFT vs. NMF-MEL-SC-RC. ............ 75
4.16 Preference test results of speech quality with 95% confidence intervals for DKPLS and NMF-MEL-SC-RC methods. Both methods used 500 exemplars/frames as training. .............................................. 76
4.17 Mean opinion scores with 95% confidence intervals for ML-GMM and NMF-MEL-SC-RC methods. .............................................. 76
4.18 Mean opinion scores with 95% confidence intervals for ML-GMM-GV and NMF-MEL-SC-RC methods. .............................................. 77
4.19 Comparison of speaker identification rates of different voice conversion approaches. .................................................. 78

5.1 Illustration of the vulnerabilities evaluation framework used in this study. 90

5.2 Diagram of a joint density Gaussian mixture model (JD-GMM) based voice conversion system. ........................................... 91

5.3 Hierarchical acoustic modelling approach for text-dependent speaker verification. .......................................................... 94

5.4 Score distributions of PLDA system before and after voice conversion spoofing. .............................................................. 103

5.5 Score distributions of the male TD-HMM system before and after spoofing. 105

5.6 Score distributions of the female TD-HMM system before and after spoofing. .............................................................. 106

5.7 Comparison across spectral distortions, FARs of TD-HMM systems and that of TD-GMM systems on the RSR2015 dataset. (a): FARs of male and female TD-HMM systems. No conversion means before spoofing attack. VC-2, VC-5, VC-10 and VC-20 represent different voice conversion attacks. (b): FARs of male and female TD-GMM systems. (c): Spectral distortions of male and female voice conversion systems. ...... 109
# List of Tables

4.1 Summary of the implemented methods and their formulations. ........ 62  
4.2 Comparison of spectral distortions of the NMF-based methods ....... 65  
4.3 Comparison of spectral distortions of the proposed and baseline methods 70  

5.1 Four groups of trial decisions in speaker verification. ............... 96  
5.2 Statistics of the genuine, impostor and impostor via voice conversion (VC) trials (subset of NIST SRE 2006 core task) for the spoofing attack study. ..................................................... 98  
5.3 Statistics of the baseline and spoofing datasets from RSR2015 database (VC=voice conversion). ............................................. 99  
5.4 A summary of text-independent (TI) and text-dependent (TD) speaker verification systems used in this study. ................................. 100  
5.5 Equal error rate results of seven different speaker verification systems before and after voice conversion attack. .......................... 102  
5.6 False acceptance rate (FAR) results of seven text-independent speaker verification systems before and after voice conversion spoofing attack. FAR is obtained by setting the decision threshold to the EER point on the baseline dataset. .............................................. 103  
5.7 Performance of text-dependent speaker verification systems under the same voice conversion spoofing attack. False acceptance rate (FAR) is obtained by setting the threshold to the equal error rate point on baseline dataset. Assuming the impostor knows the pass-phrases. ....... 104
Chapter 1

Introduction

Human speech conveys not only what a person says, but also who speaks. According to speaker recognition and speech perception research [3, 4], speaker identity is characterized by following factors,

a) *Linguistic* factor: It is relevant to the language content or the lexical message expressed by the speaker, and has the same information as presented in text format. Each speaker’s identity is characterized by his/her own lexicon, and the grammatical choice of words and sentence structure the speaker tends to use in his/her conversations.

b) *Supra-segmental* factor: This is a kind of acoustic characteristics, in particular, prosodic characteristics. It involves the duration of phonemes, syllables and words; the evolution of fundamental frequency (intonation); stress; tone.

c) *Segmental* factor: It is a type of acoustic characteristics, which relates to short-term features, such as spectrum, formants, and the shape of the glottal excitation pulse [5].

In speech processing, given fixed language content, the supra-segment and the segmental factors, also called *acoustic* factors, are the most relevant factors to speaker individuality.

This thesis focuses on voice conversion technology. It is a technique to modify one speaker’s voice to sound like it was pronounced by another speaker without changing the language content conveyed by the source speaker. Ideally, the desired technology
will be required to convert both the supra-segment and the segmental factors. It is believed that the segmental level acoustical features contain more speaker individuality information and are much easier for extraction and modeling [4]. To this end, this thesis focuses on the segmental level conversion aspect which is spectral mapping, without modifications to pitch information. What this means is that duration and intonation of the target speech will be kept the same as the source speech. Although the technique only partially solves the voice conversion issue, it remains a challenging one, and its successful implementation will still generate interesting and useful applications.

Voice conversion technology has many applications. For example, it can be used to personalize a text-to-speech (TTS) system. Current speech synthesis techniques require a large amount of speech data from a specific speaker; thus it is expensive to build a personalized TTS system. Voice conversion can be used to establish a mapping from speech signals of a source speaker (or a TTS system) to a target speaker with a limited amount of data. Voice conversion can also be used to hide a speaker’s true identity as well as to mimic a specific target speaker to attack a speaker verification system [6]. Techniques developed for voice conversion are also handy in other application, such as speech enhancement [7, 8] and speaking-aid [9, 10].

This thesis will focus on the following two aspects of voice conversion research: a) proposing new algorithms to improve voice conversion performance for realistic sounding; and b) evaluating the vulnerability of current state-of-the-art speaker verification systems using existing voice conversion techniques.

1.1 Contributions

Major issues in current spectral mapping techniques are over-fitting and over-smoothing of mapping functions. The origin of the over-fitting issue is that model parameters attempt to over-fit the sparse training data [11–13], and the resulted conversion functions have degraded mapping performance on the unseen data. The over-smoothing issue is caused by two effects: a) statistical average nature of statistical parametric approaches [14–16]; and b) the use of low-resolution features to represent spectra that leads to the loss of spectral details [17, 18]. It is reported that the over-smoothing issue will result in muffled sounding speech [14–16]. This thesis attempts to propose algorithms to alleviate the over-fitting and over-smoothing issues.
This work has three major contributions. The first contribution of this thesis is to propose a mixture of factor analyzers model to benefit from nonparallel data to train a conversion function [19], and aims to reduce the over-fitting issue. In most of the conventional voice conversion methods, only speech data from source and target speakers are involved to estimate the conversion function. In practice, there are plenty of nonparallel and high-quality speech datasets from many other speakers. The challenge is to utilize these nonparallel data to improve the voice conversion performance. To this end, a method based on factor analysis is proposed to utilize nonparallel data. It assumes that a speech feature vector can be decomposed into two components: speaker-independent phonetic and speaker-dependent speaker-specific components. The speaker-independent component is related to the language content information, while the speaker-dependent component is related to speaker individuality and is represented by a low-dimensional set of factors, also called speaker-specific vector via factor loadings. In this way, the nonparallel data are employed to estimate the speaker-independent component and factor loadings. The source and target speech data are only used to estimated the low-dimensional speaker-specific vector, and hence the number of parameters representing the speaker can be reduced considerably.

The second contribution of this thesis involves an exemplar-based sparse representation with residual compensation method as an alternative nonparametric framework for voice conversion [20, 21] attempting to reduce the over-smoothing effect. In this framework, a spectrogram is reconstructed as a weighted linear combination of speech segments, called exemplars, which span multiple consecutive spectra. The linear combination weights are constrained to be sparse to avoid over-smoothing, and the high-resolution spectra are employed in the exemplars directly without dimensionality reduction to maintain spectral details. In addition, a spectral scaling factor and a residual compensation technique are included in the framework to enhance the conversion performances, in terms of spectral distortion and speech quality.

The third contribution focuses on the use of voice conversion to attack automatic speaker verification systems [6, 22–25]. Both voice conversion and automatic speaker verification techniques are speech processing techniques dealing with speaker identity. Speaker verification aims to automatically accept or reject a claimed identity given a speech sample, while voice conversion aims to manipulate one speaker’s voice to sound like another speaker without changing language content. Hence, voice conversion can
be employed to attack speaker verification systems. In this study, a detailed vulnerabilities analysis of seven text-independent and two text-dependent speaker verification systems is conducted under spoofing attacks simulated by the mainstream joint density Gaussian mixture model voice conversion.

The first work on the mixture of factor analyzers discussed in Chapter 3 is published in IEEE Signal Processing Letters [19]. The second work on exemplar-based sparse repertation is discussed in details in Chapter 4 and published in the conference: the 8th ISCA Speech Synthesis Workshop (SSW8) [20], and its extended version is published in the journal: IEEE/ACM Transactions on Audio, Speech and Language Processing [21]. The third work on vulnerabilities analysis of speaker verification under voice conversion attacks is discussed in details in Chapter 5 and published in the conference: IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP) 2012 [22], Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC) 2012 and 2013 [6, 23], Interspeech 2013 [24], and the extended version has been submitted to: IEEE Transactions on Audio, Speech and Language Processing [25].

1.2 Thesis outline

The remainder of this thesis is organized as follows:

Chapter 2 provides an overview of the state-of-the-art voice conversion techniques, which serve as the foundation of the studies in this thesis.

Chapter 3 proposes a mixture of factor analyzers method which integrates prior knowledge from nonparallel speech into the training of conversion function.

Chapter 4 proposes an exemplar-based sparse representation with residual compensation to deal with high-resolution spectral feature mapping for voice conversion.

Chapter 5 analyzes the vulnerabilities of speaker verification under voice conversion attack to link the two techniques from the perspective of dealing with speaker identity.

Chapter 6 concludes this thesis and provides possible future research directions.
Chapter 2

Background

This chapter presents the background knowledge of voice conversion and reviews the current state-of-the-art spectral conversion techniques. It provides a background to describe the contributions of chapters 3, 4 and 5. The chapter is structured as follows. Section 2.1 briefly introduces the speech production process and the speaker identity perception; Section 2.2 presents an overview of voice conversion; Section 2.3 reviews the current state-of-the-art spectral mapping techniques; Section 2.4 discusses the application of voice conversion; and Section 2.5 presents a summary of the chapter.

2.1 Speech production and speaker identity

Speech production is the process that turns the intent of uttering a sentence to an audible speech signal [26]. Figure 2.1 illustrates the human organs associated with human speech production process. During the speech production process, the diaphragm first forces air from the lung to the trachea; the airflow then moves from the trachea to the larynx and into the vocal tract to generate the audible speech signal. When the air flows through the larynx and reaches the glottis, if the vocal folds vibrate, the airflow will become a voicing sound source, which contains a set of different frequencies called harmonics; if the vocal folds do not vibrate, the air flows freely to result in an unvoiced sound source like random noise. In terms of modeling speech production in the source-filter model [27], the source signal corresponds to the airflow coming from the glottis, while the vocal tract which can be thought as a filter consists of the pharynx,
Chapter 2. Background

the nasal and the oral cavities, and serves to modify the sound source to generate a large variety of sounds. These sounds can be grouped as voiced and unvoiced speech.

Figure 2.1: Illustration of the human organs associated with the human speech production process [1].

A voiced speech is produced when the vocal folds vibrate during speech production. Figure 2.2 shows the waveform (left figure) and spectrum (right figure) representations of a voiced speech, in particular the phoneme /iː/. The waveform representation clearly shows the interval of a periodic signal; the period is due to the rate of opening and closing of the glottal pulse. The inverse of this period is known as the fundamental frequency (F0) of the signal. F0 is the frequency of the vocal fold vibrations and correlates with the changes in vocal fold tension and subglottal air pressure. The right figure of 2.2 shows frequency representation of the signal. The spectral shape and the spectral envelope allow human to identify different sounds. The peaks of the spectral envelope are known as formants. Usually, the first four formants, namely F1-F4, are strong enough to show up in the spectral envelope and are more meaningful in speech processing [27]. These information, F0 and formants, are all related to speaker identity.

An unvoiced speech is produced by the relaxed layrnx where air flows freely across.
Figure 2.2: A voiced sound example of the phoneme /i:/, the signal of which is periodic. The left figure is the time-domain representation of the speech signal, also known as waveform representation. The right figure is the frequency-domain representation of the same speech signal, known as spectrum and corresponding spectral envelope.

Figure 2.3 shows the waveform (left figure) and spectrum (right figure) representations, respectively, of an unvoiced speech, that is phoneme /s/. From the waveform representation, the signal resembles a random noise without periodicities. Since there is no fundamental frequency for an unvoiced speech, the spectrum does not show harmonics. From the perspective of the spectrum representation, the formant structure is obviously different from that of the phoneme /i:/.

As discussed above, a speaker controls the position of articulators and the glottis to produce a specific language content. To produce speech with the same phonetic meaning, human articulatory organs are in general in similar positions. However, as the physical oral cavity, the nasal cavity and the glottis of each individual are slightly different from other speakers, different speakers have slightly different speech timbre and this characterizes each speaker’s voice.

Figure 2.4 presents a comparison of a male and a female speech signals with the same language content. Although the two speakers speak the same utterance, their speech signals are clearly different. From top to bottom, there are several layers of information in Figure 2.4: (a) the time-domain waveform representation; (b) frequency-domain spectrogram representation corresponding to (a); (c) the first four, F1-F4, formant tracks; (d) the fundamental frequency contours; and (e) intensity trajectories.
Figure 2.3: An unvoiced sound example of the phoneme /s/. The left figure is the waveform which looks like random noise, and the right figure is the corresponding spectrum representation.

The variations of spectrogram in the time domain are due to the positions of the articulators. The first four formant tracks are also different between the two speakers. The F0 of the female is higher than that of the male speaker. F0 of a typical adult male is in the range from 85 to 185 Hz, and that of a typical adult female is from 165 to 255 Hz [28]. The prosodic features composed of the intonation patterns, the intensities and the duration of each phoneme are also distinguishable between the male and the female speech. The intonation pattern represents the fundamental frequencies over a longer time, and the prosody reflects various suprasegmental features, which are not confined to any short-term units.

From the perspective of speech perception, pitch and timbre characteristics, which relate to fundamental frequency and formants, respectively, carry important information to characterise a speaker’s identity, and these characteristics could be used to distinguish speakers [29]. In particular, vowels, voiced consonants and nasals contain significant speaker characteristics in perceptual speaker identification [30–33]. Similar to human speaker recognition, speech processing technologies that relate to speaker individuality, such as voice conversion and speaker verification, also confirm the effectiveness of these features [4, 34–36].
Figure 2.4: Comparison of male (left) and female (right) speech signals to illustrate the difference of speech representations across speakers. (a) Time-domain waveform representation; (b) Frequency-domain spectrogram representation; (c) The first four formant tracks, F1-F4; (d) Fundamental frequency trajectories; (e) Intensity trajectories. These layers of information are aligned with corresponding phonemes. Praat [2] is used to create the figure.
2.2 Overview of voice conversion

The goal of voice conversion is to modify a source speaker’s speech signal to sound like it was produced by a target speaker without changing the language content. In other words, a voice conversion system only modifies speaker-dependent characteristics of the speech signal such as spectral shape, formants, F0, intonation, intensity and duration to change the speaker identity while keeping the speaker-independent information. The primary module of a voice conversion system is the conversion function whose task is to modify the source speaker’s features to a target speaker. The conversion function can be formulated as follows:

\[ y = \mathcal{F}(x), \]  

(2.1)

where \( x \in \mathbb{R}^{d \times 1} \) and \( y \in \mathbb{R}^{d \times 1} \) are the source and target features, respectively, \( d \) is the dimensionality of features, and \( \mathcal{F}(\cdot) \) is the conversion function to map source features to target features.

Figure 2.5 illustrates a typical voice conversion framework consisting of two stages: offline training and runtime conversion. At the offline training stage, parallel training data are first prepared for a source speaker \( X \) and a target speaker \( Y \). These data are then analyzed by a speech production model [37–39] to generate independent components such as spectrum, excitation and F0. These components are further passed to a feature extraction module to extract low-dimensional features such as Mel-cepstral coefficients (MCCs) or linear predictive cepstral coefficients (LPCCs). For each parallel utterance, frame alignment techniques such as dynamic time warping (DTW) are employed to pair up the source and target features,

\[ X = [x_1, x_2, \ldots, x_n, \ldots, x_N] \]  

(2.2)

\[ Y = [y_1, y_2, \ldots, y_n, \ldots, y_N], \]  

(2.3)

where \( X \in \mathbb{R}^{d \times N} \) and \( Y \in \mathbb{R}^{d \times N} \) are the paired source and target data, respectively, \( N \) is the number of frames, and \( x_n \) and \( y_n \) are the paired source and target features at the frame index \( n \), respectively. Finally, the parallel data \( Z = [X; Y] \), where each column \( z_n = [x_n; y_n] \in \mathbb{R}^{2d} \) is a joint vector, are employed to estimate the conversion function \( \mathcal{F}(\cdot) \).
At runtime, a source speech signal is passed through the same speech analysis and feature extraction modules to generate the low-dimensional features $x$. The conversion function $F(\cdot)$ estimated during training is applied to generate the target speaker’s features $\hat{y} = F(x)$. Finally, these generated features $\hat{y}$ are passed to a synthesis filter to reconstruct an audible target speaker’s speech signal. The synthesis filter is also named as a speech reconstruction module.

Each module in the framework faces its own challenges. This section provides a general overview of the typical approaches in each module.
2.2.1 Speech analysis and reconstruction

The speech analysis and reconstruction (also called synthesis filter in Figure 2.5) modules are two important and necessary modules for a voice conversion system. As shown in Figure 2.5, the speech analysis module operates on the input speech signals and decomposes each speech signal into several independent components for flexible modeling and modifications, while the speech reconstruction module which is usually realized in the form of a synthesis filter operates on the modified features to generate an audible speech signal.

The two modules have a strong relationship in a voice conversion system, and are usually based on speech production models, such as the source-filter model. In practice, the two modules should be provided in pairs. When choosing the speech production model, the speech analysis module is expected to be able to separate a speech signal into several mutually independent representation components, such as spectrums, fundamental frequencies (F0) and excitations, for flexible modifications without introducing artifacts. This requirement enables the representation components to be modified independently.

The speech reconstruction module is supposed to able to synthesize high quality speech from the modified speech representations. This requirement influences the quality of the reconstructed speech signals. In practice, both the speech analysis and speech reconstruction modules are not perfect, and artifacts will still occur in the converted speech.

The most commonly used speech production models for designing the speech analysis and the speech reconstruction modules in voice conversion research are briefly discussed here:

**Pitch synchronous overlap and add (PSOLA)** is a technique based on the decomposition of a speech signal into several overlapping speech segments [40]. Each speech segment represents one of the successive pitch periods of the speech signal, and the sum (overlap and add) of these speech segments can be utilised to regenerate the speech signal. As PSOLA operates directly on the waveform of the speech signal without any speech production model, the analysis and reconstruction do not lose any details of the speech signal. Although PSOLA techniques have been adopted for voice conversion in [41–43], modification of signals with
PSOLA is not flexible, as the raw waveform is not flexible for modifications.

**STRAIGHT** which stands for “Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum” is based on the source-filter theory [37]. STRAIGHT is a high-quality vocoder, which decomposes the speech signal into three components: a) a smooth spectrogram which is free from periodicity in time and frequency; b) a fundamental frequency (F0) contour which is estimated using a fixed-point algorithm; and c) a time-frequency periodicity map which captures the spectral shape of the noise and also its temporal envelope. STRAIGHT allows flexible speech manipulation, as the three components are mutually independent. It has been widely used in voice conversion applications [12, 34, 44–46].

**Linear prediction (LP) model** is an all-pole filter model based on the idea that a speech sequence can be modeled as a linear combination of its previous \( p \) speech samples, where \( p \) is the order of LP coefficients [38, 47, 48]. A speech signal \( s[n] \) can be expressed as:

\[
s[n] = \sum_{k=1}^{p} a_k s[n - k] + e[n],
\]  

(2.4)

where \( a_k \) are the linear predictor coefficients, and \( e[n] \) is the prediction error. By taking Z-transform of Eq. (2.4), we have

\[
S(z) = X(z) \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = X(z) \frac{1}{A_p(z)},
\]  

(2.5)

where \( S(z) \) and \( X(z) \) are the Z-transforms of the speech and excitation signals, respectively, and the all-pole transfer function \( \frac{1}{A_p(z)} \) is known as the synthesis filter. The linear prediction model has been used for voice conversion applications in [49–52].

**Harmonic plus noise model** assumes that a speech signal is represented as a harmonic component plus a noise component that is separated in the frequency domain by a cut-off frequency, referred as the maximum voiced frequency [39]. The harmonic component is modeled as the sum of harmonic sinusoids up to the maximum voiced frequency, while the noise component is modeled as Gaussian
noise filtered by a time-varying autoregressive filter. The decomposition of a
speech signal into harmonic and noise components allows for flexible modifi-
cations. The harmonic plus noise model has been employed for voice conversion
applications in [53–55].

Due to the flexibility and good performance, the STRAIGHT vocoder is adopted
in this thesis.

2.2.2 Feature extraction

As presented in Figure 2.5, a feature extraction module uses the output of the speech
analysis module as input to extract feature representations. The output of the speech
analysis module usually contains spectral and prosodic components. The dimension-
ality of prosodic representations, such as F0, intonation, duration and intensity, is
usually low, and the prosodic features are easy for modeling. However, spectral repre-
sentations, such spectral envelope, contains significant amount of information, and the
dimensionality is usually considerably higher. To robustly model the spectral represen-
tation, the objective of feature extraction is to exploit representative low-dimensional
representations from the raw high-dimensional spectra. In this thesis, the focus is on
spectral mapping, and hence only spectral feature extraction is discussed further in
this section.

The spectral features are extracted by following the criteria as:

a) be able to represent the speaker individuality well;

b) fitting the spectral envelope well and is able to be converted back to spectral
envelope;

c) having good interpolation properties and allowing for flexible modification.

Features with the above properties will be suitable for statistical parametric modeling.
Some popular feature with such properties are briefly discussed below:

**Cepstral Coefficients** have been widely used in speech processing technologies, such
as speech recognition, speaker recognition and speech synthesis. They model
both spectral peaks and spectral valleys, and provide good envelope fit which
is important for synthesis. Mel-Cepstral Coefficients (MCCs), generalized Mel-Cepstral Coefficients (GMCCs), and Linear Predictive Cepstral Coefficients (LPCC) are popular cepstral features used for voice conversion applications [8, 12, 34, 56].

**Line Spectral Frequency (LSF)** parameters have good quantization, interpolation properties and good representation of the formant structure [57, 58]. As such, they have been successfully applied to speech coding and speech synthesis. In the text-to-speech research community, better performance was achieved using LSF features over mel-cepstral coefficients and other features. However, there are issues with LSFs in voice conversion. If the conversion function is not stable, it is hard to guarantee the converted LSFs to be within the range of $0 - \pi$ and in ascending order. This will cause the synthesis filter to be unstable. LSF has been widely used in voice conversion [52, 54, 55, 59, 60].

Other features such as formants are also employed in voice conversion [61, 62], but it is a challenging task to accurately extract formants from speech signals.

It is noted that in some voice conversion systems [20, 21], the feature extraction module is skipped, as some statistical models are powerful enough to handle the relative high-dimensional raw spectra. However, such kinds of voice conversion systems usually require more computational costs.

### 2.2.3 Frame alignment

The frame alignment module as illustrated in Figure 2.5 uses the output of the feature extraction module as input, and the objective of frame alignment is to pair up source and target features that have the same language content. In this way, the conversion function operate only on the speaker characteristics without changing the linguistic aspect.

There are two kinds of training data for voice conversion, parallel and nonparallel data. Parallel data mean that the source and target speakers utter the same sentences during recording, implying cooperative users, while nonparallel data do not have such constraint, allowing the speakers to speak freely. Accordingly, there are two kinds of alignment techniques, parallel and nonparallel alignment, to operate on the corresponding data.
Parallel alignment

When parallel data are available, dynamic time warping (DTW) \[63\] is one of the most popular techniques to perform frame alignment. DTW aligns the source and the target feature sequences by minimizing the spectral distance with temporal structure constraint. In practice, voice activity detection and linguistic information can also be used as constraints to improve the alignment performance. A detailed comparison of several different implementations of DTW is presented in \[64\].

Nonparallel alignment

When parallel data are not provided, DTW is not applicable. Several nonparallel frame alignment techniques have been proposed to address this issue. One of the simplest techniques is the frame-based alignment method, where each source or target feature is paired up with the nearest target or source feature by minimizing the Euclidean distance. This approach does not require any linguistic or contextual information, and has been used as the reference baseline in several studies \[65, 66\].

In \[65\], a class-based frame alignment method was proposed. In this method, the source and target features are first clustered independently into the same number of classes. After the mapping between the source and the target classes is established, each source feature is paired with the nearest target feature in the corresponding class by minimizing the Euclidean distance. The advantage of class-based method against frame-based method is that an artificial phonetic cluster is learned first within each speaker and then the mapping is performed between corresponding clusters. The use of an artificial phonetic cluster can reduce the searching space to computational efficiency, and also improves the alignment accuracy.

In \[67\], a HMM-based speaker-independent speech recognizer was used to label all the source and target speech frame with an HMM state identity. In this approach, given a state sequence of the source speech, the longest matching sequence is found from the target speech. Such a process is repeated until the whole source sequence is paired up with a target sub-sequence. In this approach, contextual information is included. However, the performance of speech recognizer will affect the alignment accuracy considerably. In addition, the acoustic distance between source and target features is not directly exploited.
In [66], an iterative updated frame alignment was proposed. In this method, an auxiliary transformation function is established to produce pseudo parallel data given nonparallel data. In this approach, the auxiliary transformation function and the pseudo parallel data are updated iteratively until a pre-defined convergence requirement is satisfied. The advantage of this method is the case that no additional knowledge is required, and hence this method is applicable for cross-lingual voice conversion applications.

2.2.4 Prosody conversion

Typically, voice conversion involves spectral mapping and prosody conversion. Spectral mapping seeks to change the voice timbre, while prosody conversion seeks to modify the prosodic features, such as fundamental frequency, intonation and duration. This subsection reviews the prosody conversion techniques, and Section 2.3 presents a detailed review of the spectral mapping techniques.

The most common technique is to change the pitch of the speaker by normalizing the mean and variance of the source speaker’s (log-)F0 distribution to the target speaker’s mean and variance. This can be implemented by a straightforward linear transformation at the frame-level (or instantaneous) F0 values [68]. Specifically, the converted F0 value of a single frame \( x' \) is obtained by

\[
x' = \frac{\sigma_y}{\sigma_x} (x - \mu_x) + \mu_y,
\]

where \( x \) is the source speaker’s F0 value, \( \mu_x \) and \( \mu_y \) are the means and \( \sigma_x \) and \( \sigma_y \) are the standard deviations of the the source and the target speakers’ training data, respectively. This method only changes the global F0 level and dynamic range while retaining the shape of the source contour.

Extensions of this approach, but still operating on instantaneous F0 values, include methods such as higher-order polynomial [69], GMM-based mapping [70] and piecewise linear transformation based on hand-labelled intonational target points [71]. Transformation methods for the instantaneous F0 are simple and work well for speakers with “similar” intonation. For speakers with drastically different intonation patterns, however, it might be advantageous to convert the F0 contours (intonation contours)
instead [35, 36, 69, 70]. In these methods, the prosodic segments (e.g. syllables or entire utterances) are represented either as variable-length sequences processed with dynamic time warping (DTW) [69] or, alternatively, by parameterizing each prosodic segment by a fixed-dimensional parameter vector [35, 36] which is computationally more feasible. For an extensive objective and subjective comparison of five different F0 conversion methods, including both instantaneous and contour-based methods, refer to [70].

In [72], a text-independent F0 conversion method is proposed. The dynamic feature is included to generate smooth F0 trajectories. In addition, a histogram equalization technique [73] is applied to the generated F0 trajectories to avoid over-smoothing. Both objective and subjective evaluation confirmed the effectiveness of this method.

Even though the contour-based conversion may outperform the instantaneous conversion methods [70], care must be taken: since the intonation contour depends on both lexical factors (e.g. interrogative vs declarative sentence) and various paralinguistic factors (e.g. language and speaker’s mood), it is difficult to isolate only the speaker-dependent component for conversion purposes. Consequently, if the training data and the utterance under conversion do not match in the lexical and paralinguistic attributes, the converted utterance is expected to sound unnatural. Additionally, some of the methods require syllable-level annotation, and, importantly, most of them require a parallel training corpus. Note that this is not the case for the baseline mean and variance conversion method which enjoys complete text-independency. The only prosody conversion not relying on parallel data that we are aware of is [36]. In that study, the authors used syllable-level F0 and duration features in a maximum likelihood linear regression (MLLR) conversion method.

### 2.3 Spectral mapping techniques

In this section, several popular spectral conversion methods related to this thesis are discussed. The shortcomings of the spectral conversion methods are also highlighted to motivate the contributions of this work in Chapters 3 and 4.
2.3.1 Vector quantization (VQ)

A straightforward approach to perform spectral mapping is to use vector quantization (VQ) [74]. The simplest way to implement VQ for voice conversion is illustrated in Figure 2.6. During offline training phase, the aligned source-target feature pairs as presented in Eqs. (2.2) and (2.3) are stacked to form the joint vectors $Z$. A mapping codebook is then built using the joint vectors. As illustrated in Figure 2.6, the square points represent source entries in the codebook while the circle points stand for target entries in the codebook. At runtime, each source feature is paired up with one source feature entry in the codebook, and the corresponding target feature entry is selected as the converted feature.

![Figure 2.6: Illustration of joint vector quantization. The square points are entries for source feature codebook while circle points stand for entries for target feature codebook. Note that the square points and circle points always appear in pairs to stand for joint vectors.](image)

Different from the approach illustrated in Figure 2.6, a classical implementation was proposed in [74] to have the source and target feature vectors quantized separately. A dynamic time warping technique is then used to find the correspondences, which are accumulated to make histograms. The histograms are then employed as weighting
functions to produce the mapping codebook which is defined as a linear combination of the target speaker’s vectors.

The advantage of using the VQ method is that the selected target feature keeps the spectral details as original one, however, the hard clustering nature of VQ results in the incoherence phenomenon across frames [34,59,68].

2.3.2 Gaussian mixture model based methods

To avoid the hard clustering nature of VQ, Gaussian mixture model (GMM) based methods using soft clustering were proposed in [34,59,68]. Among the statistical parametric methods to model the conversion function, joint density Gaussian mixture model (JD-GMM) method [34,59] is one of the most successful methods due to its probabilistic treatment and flexible implementation. By using different criteria to optimize the conversion function, JD-GMM has different implementations. This subsection introduces the minimum mean square error based solution [59] and the maximum likelihood trajectory generation solution [34].

**Minimum mean square error based solution**

The following describes the JD-GMM implementation using minimum mean square error criteria. The JD-GMM method involves two phases: offline training and runtime conversion phases. During the training phase, Gaussian mixture model (GMM) is adopted to model the distribution of the paired feature sequence $Z$, which represents the joint distribution of source speech $X$ and target speech $Y$. The joint probability density is given as follows:

$$P(X, Y) = P(Z) = \sum_{k=1}^{K} w_k^{(z)} \mathcal{N}(z | \mu_k^{(z)}, \Sigma_k^{(z)}),$$

$$\mu_k^{(z)} = \begin{bmatrix} \mu_k^{(x)} \\ \mu_k^{(y)} \end{bmatrix}, \Sigma_k^{(z)} = \begin{bmatrix} \Sigma_k^{(xx)} & \Sigma_k^{(xy)} \\ \Sigma_k^{(yx)} & \Sigma_k^{(yy)} \end{bmatrix},$$

where $K$ is the number of Gaussian components, $\mu_k^{(z)}$ and $\Sigma_k^{(z)}$ are the mean vector and the covariance matrix of the $k$th Gaussian component $\mathcal{N}(z | \mu_k^{(z)}, \Sigma_k^{(z)})$, respectively. The prior probability $w_k^{(z)}$ of the $k$th Gaussian component is constrained by $\sum_{k=1}^{K} w_k^{(z)} = 1.$
To estimate the model parameters $\lambda(\varepsilon) = \{w_k^{(\varepsilon)}, \mu_k^{(\varepsilon)}, \Sigma_k^{(\varepsilon)} | k = 1, 2, \ldots, K\}$, the well-known expectation-maximization (EM) algorithm is adopted to maximize the likelihood of the training data.

During the runtime conversion phase, the JD-GMM model is employed to implement the conversion function. Specifically, for each input source vector $x$, the conversion function $F(x)$ realized using the minimum mean square error is employed to predict the target’s feature vector $\hat{y}$ by [59]:

$$\hat{y} = F(x) = \sum_{k=1}^{K} p_k(x)(\mu_k^{(y)} + \Sigma_k^{(yx)} (\Sigma_k^{(xx)})^{-1} (x - \mu_k^{(x)})),$$  \hspace{1cm} (2.8)

$$p_k(x) = \frac{w_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \Sigma_k)},$$

where $p_k(x)$ is the posterior probability of the source vector $x$ generated from the $k^{\text{th}}$ Gaussian component.

**Maximum likelihood estimation of parameter trajectory using dynamic features**

The following then describes the GMM-based voice conversion using the maximum likelihood trajectory generation approach. Similar to the minimum mean square error solution of the JD-GMM described in the previous section, the DTW algorithm is first employed to align the source $X$ and target $Y$ features. To include temporal information, dynamic feature is included. Hence, the source and target features at frame $t$ are represented as $X_t = [x_t^\top, \Delta x_t^\top]^\top \in \mathcal{R}^{2d}$ and $Y_t = [y_t^\top, \Delta y_t^\top]^\top \in \mathcal{R}^{2d}$, respectively. $x_t^\top$ and $\Delta x_t^\top$ are the static and dynamic source features, respectively, and $y_t^\top$ and $\Delta y_t^\top$ are the static and dynamic target features, respectively. In this way, the joint vector $Z_t$ is written as $Z_t = [X_t^\top, Y_t^\top]^\top$. Using the parallel joint vectors, conventional JD-GMM training framework can be employed as for Eq. (2.7).

At runtime conversion, to generate the target feature trajectory $\hat{Y}^{(s)}$, we maximize the following likelihood function:

$$\hat{Y}^{(s)} = \arg \max P(Y|X, \lambda(Z)) = \arg \max \sum_{\text{all } m} P(m|X, \lambda(Z)) P(Y|X, m, \lambda(Z)), \hspace{1cm} (2.9)$$
where \( m = \{m_1, m_2, \ldots, m_T\} \) is the Gaussian mixture sequence, and \( \lambda^{(Z)} \) is the model parameters set. Note that the feature vectors \( Y \) which include static and dynamic features have the following linear relationship with the static feature vector sequence \( Y^{(s)} \):

\[
Y = W Y^{(s)},
\]  

(2.10)

where \( W \) is the \( 2dT \)-by-\( dT \) matrix as detailed in [34].

As discussed in [34], the Gaussian mixture sequence in Eq. (2.9) is latent variable. Therefore, EM algorithm can be employed to generate the parameter trajectory. In practice, we can approximate the suboptimum Gaussian mixture sequence with degrading the performance. The approximated likelihood function with a optimum mixture sequence is given as follows:

\[
P(Y|X, \lambda^{(Z)}) \approx P(\hat{m}|X, \lambda^{(Z)}) P(Y|X, \hat{m}, \lambda^{(Z)}).
\]  

(2.11)

In Eq. (2.11), the suboptimum Gaussian mixture sequence \( \hat{m} \) is determined by

\[
\hat{m} = \arg \max P(\hat{m}|X, \lambda^{(Z)})
\]

Hence, we have the following objective function:

\[
\mathcal{L} = \log P(Y|X, \lambda^{(Z)}) \approx \log P(\hat{m}|X, \lambda^{(Z)}) P(Y|X, \hat{m}, \lambda^{(Z)}).
\]  

(2.12)

To generate the parameter trajectory \( \hat{y} \), we maximize the the log-likelihood \( \mathcal{L} \) with Eq. (2.10) constraint. Thus, we have

\[
\hat{y} = (W^T D^{(Y)}_{\hat{m}})^{-1} W^{-1} W^T D^{(Y)}_{\hat{m}}^{-1} E^{(Y)}_{\hat{m}},
\]  

(2.13)

where

\[
E^{(Y)}_{\hat{m}} = [E^{(Y)}_{\hat{m}_1,1}, E^{(Y)}_{\hat{m}_2,2}, \ldots, E^{(Y)}_{\hat{m}_T,T}],
\]

\[
D^{(Y)}_{\hat{m}}^{-1} = \text{diag}[D^{(Y)}_{\hat{m}_1}^{-1}, D^{(Y)}_{\hat{m}_2}^{-1}, \ldots, D^{(Y)}_{\hat{m}_T}^{-1}],
\]

\[
E^{(Y)}_{\hat{m}_t} = \mu^{(Y)}_{\hat{m}_t} + \Sigma^{YX}_{\hat{m}_t} \Sigma^{XX}_{\hat{m}_t}^{-1} (X_t - \mu^{(X)}_{\hat{m}_t}),
\]
and
\[ D_{\hat{m}_t} = \Sigma_{\hat{m}_t}^{YY} - \Sigma_{\hat{m}_t}^{YX} \Sigma_{XX}^{-1} \Sigma_{XX}^{XY}. \]

We note that \( E_{\hat{m}_t,b}^{(Y)} \) has the similar form as that in the conversion of minimum mean square error method.

Many research work have shown that maximum likelihood estimation of parameter trajectory methods have better speech quality and lower spectral distortion than minimum mean square error method [34, 45]. Therefore, the maximum likelihood estimation of parameter trajectory method is currently the established baseline system for voice conversion research.

**Issues in GMM-based methods**

We note that during the JD-GMM model parameter estimation process, the mean vector of each Gaussian component is updated as:

\[ \mu_{k}^{(z)} = \frac{\sum_{t=1}^{T} z_{t} p_{k}(z_{t}, \lambda(z))}{\sum_{t=1}^{T} p_{k}(z_{t}, \lambda(z))}. \]  

(2.14)

Similarly, the covariance matrix of each Gaussian component is updated as:

\[ \Sigma_{k}^{(z)} = \frac{\sum_{t=1}^{T} p_{k}(z_{t}, \lambda(z))(z_{t} - \mu_{k}^{(z)})(z_{t} - \mu_{k}^{(z)})^\top}{\sum_{t=1}^{T} p_{k}(z_{t}, \lambda(z))}. \]  

(2.15)

From (4.17) and (4.18), we observe that when calculating mean and covariance for each Gaussian component, all the training samples are used, which is the so-called statistical average. The statistical average results in over-smoothing of the converted speech. We also note that if the correlation between the paired source and target feature vectors is low, the value of the covariance matrix \( \Sigma_{k}^{(yz)} \) will be very small. As illustrated in Fig. 2.7, the covariance values of the non-diagonal elements and the cross covariance values are very small. The elements in \( \Sigma_{XX}^{XY} \Sigma_{XX}^{-1} \) will be very small, therefore, only \( \mu_{k}^{(y)} \) contributes to the converted speech as observed and reported in [15].

To address the over-smoothing [14–16] and over-fitting [11–13] problems of JD-GMM, a number of methods have been proposed. In [34], global variance has been
Figure 2.7: An example of the covariance matrix from Gaussian mixture model.

included in the spectral trajectory generation process to overcome the over-smoothing problem. Different from [34], in [75], global variance has been included in the training of Gaussian mixture model to reduce over-smoothing. In [12], partial least square regression method has been proposed to avoid over-fitting problem when the parallel training data is limited, and local linear transformation method is implemented by using nearby training data not all the training data to estimate the transformation function in [14]. In [76], bag-of-Gaussian model is proposed to avoid the over-smoothing problem by increasing the number of local linear transformation. In [77], mutual information criterion has been used during parameter generation to enhance the dependency between the source and converted feature vectors. Discriminative training has been employed for GMM training for alleviating the over-smoothing problem [78]. In addition, noisy
channel model [46] methods have been proposed to utilize additional data to improve the performance of the conversion function.

### 2.3.3 Partial least squares regression based methods

As discussed above, when only a small amount of training data is available, it is hard to estimate the full covariance matrices in the Gaussian mixture model, especially when the number of features for training is less than the number of parameters in the model. To avoid estimating the full covariance matrices, cross diagonal covariance matrices are employed as a simplified alternative implementation [11]. A cross diagonal covariance matrix only has non-zero diagonal and cross-covariance elements as illustrated in Figure 2.8. This simplification leads to convert each source feature independently from the others, as a result leads to degraded quality of the converted speech.

![Cross diagonal matrix diagram](image.png)

**Figure 2.8:** An example of a cross diagonal matrix, in which only diagonal and cross diagonal elements are non-zeros.

To respond to the problem, a partial least squares (PLS) regression based conversion approach was proposed in [12]. This approach assumes that the source and the target data can be projected into a latent space, in which the relationship between source and target features is established. With such an idea, the following decompositions
are hence performed on the parallel data.

\[ X \approx OP, \]  
\[ Y \approx VQ, \]

where \( X \in \mathcal{R}^{d \times N} \) and \( Y \in \mathcal{R}^{d \times N} \) are the source and the target training data, respectively, \( P \in \mathcal{R}^{p \times N} \) and \( Q \in \mathcal{R}^{p \times N} \) are projections or factor matrices of \( X \) and \( Y \), respectively, and \( O \in \mathcal{R}^{d \times p} \) and \( V \in \mathcal{R}^{d \times p} \) are loading matrices corresponding to \( P \) and \( Q \), respectively.

By solving Eqs. (2.16) and (2.17), a transformation/regression matrix \( T \) can be found to perform feature transformation to generate the target speech.

\[ \hat{Y} = TX, \]

where \( T \in \mathcal{R}^{d \times d} \) is the regression matrix, and \( \hat{Y} \) is the generated target features. As \( T \) is formed from low-rank matrices which have less parameters, PLS regression is hence efficient when limited parallel training data are available. The experimental results reported on CMU ARCTIC confirmed the effectiveness of the PLS regression based method over the conventional JD-GMM method.

The PLS regression based conversion method converts each frame of speech independently from the neighboring frames, without taking temporal information constraint into account. Moreover, it assumes a linear relationship between the source and the target features.

To address those issues, a dynamic kernel partial least squares (DKPLS) regression approach was proposed in [45]. This approach assume the source and the target features have a nonlinear relationship. The nonlinearity is implemented by a Gaussian kernel function, which is used to project source features into a high-dimensional kernel space. In addition, to model the time dependencies in the feature sequence, kernel vectors of three consecutive frames are stacked as a supervector, and this stacked supervector is employed as the source feature. The standard partial least squares regression is then employed to model the relationship between the supervectors and the target features. The conversion process of this approach is illustrated in Figure 2.9.

Experiments were conducted on the VOICES database. The experimental results
showed the effectiveness of the DKPLS method in comparison with a large set of baseline methods, including joint-density Gaussian mixture model with dynamic feature constraint method and several variant methods of basic PLS method proposed in [12].

Although DKPLS works effectively in modeling the relationship between the source and the target features, there are three major drawbacks. First, the temporal information in the target feature sequence is still ignored. One possible extension to this approach is to include dynamic features for the target features, and use such dynamic features as a constraint to generate the target feature trajectory. Second, it is not flexible in modeling the long-term temporal information. The DKPLS approach stacks consecutive kernel vectors into a supervector. If long-term temporal information is included, the dimensionality of the supervector will be considerably high, as a result the PLS regression may not be robust to handle such high-dimensional supervectors. Last, DKPLS uses low-resolution mel-cepstral coefficients (MCCs) in the source and target features, the performance of using high-resolution spectra is still unknown.

### 2.3.4 Neural network based methods

Artificial neural network (ANN) is a statistical model inspired by human brain. It is able to approximate nonlinear relationships between input and output observations [79]. In the context of voice conversion, several attempts have been made to map the spectral features.

In [61], a feed-forward neural network was trained using the back-propagation al-
algorithm to transform the formants of a source speaker to those of a target speaker, and a formant vocoder was employed to synthesize an audible speech signal. In the implementation, the first three formants were used to represent the vocal tract shapes. However, no objective and subjective results were reported to examine the effectiveness of the method. It is noted that the formant extraction accuracy will also affect the performance of the formant-based conversion system.

Other than mapping formants, in [80], a three-layer radial basis function (RBF) network with Gaussian basis was used to transform the vocal tract characteristics, which were represented by linear predictive coding (LPC) based spectral envelopes. However, the experiments only provided some simulation results of synthesizing five vowels. In addition, no baseline approaches were provided as reference, and it is hard to tell the effectiveness of the method comparing with the mainstream methods.

The mapping ability of artificial neural networks for voice conversion was systematically assessed in [56]. In the study, a four-layered feed-forward neural network was designed to transform the mel-cepstral coefficients (MCCs) of a source speaker to those of a target speaker. During offline training, a generalized back-propagation learning [61] was adopted to estimate the weights of the neural network by minimizing the mean square error between the source MCC features and the target MCC features, and the weights were used to represent the mapping function between source and target features. At runtime, the conversion was straightforward. Source features were passed into the neural network as input, and the estimated weights were applied to transform the features layer-by-layer until the top layer was reached. Thus, the feature in the output layer was generated as the converted features. The experimental results showed that ANN-based spectral mapping yields results which are as good as that of the GMM-based method. This work also shows the advantage of ANN, that is to use multiple source frames to predict a target frame. GMM-based methods are not able to do this due to the high dimensionality.

A restricted Boltzmann machine (RBM) was employed to model the joint distribution of the source and target spectral features in [81]. RBM is a probability density model [82] which can capture the inter-dimensional and inter-speaker correlations through the joint spectral features. During training, a RBM was trained using the joint spectral features under maximum likelihood criterion. At runtime, the target features were generated by maximizing the conditional output probability given source
features. Experimental results showed that the RBM conversion method significantly outperform the joint-density Gaussian mixture model (JD-GMM) method. The experiments also showed that the RBM was able to handle high-resolution spectra directly without dimensionality reduction, which is required by the JD-GMM based methods.

Inspired by the success of deep learning techniques in speech recognition [83,84], deep belief nets (DBNs) were employed in [85] to perform voice conversion. DBNs have a deep architecture that is able to automatically capture the underline relationship between source and target features. During training, two DBNs are trained for source and target speakers independently, after that, a neural network (NN) is employed to concatenate the two DBNs. The weight parameter of NN is estimated by minimizing the errors between the outputs of the two DBNs. Finally, the whole network includes two DBNs and one NN is fine-tuned by the back-propagation algorithm. The experimental results confirmed the effectiveness of the DBN-based method over the classic JD-GMM based method.

The recent work inspired by deep learning techniques presents positive results over the mainstream GMM-based method [81,85]. However, the interpretation of the neural network is not as good as other statistical models and neural network-based methods perform the transformation like a “black box”. In addition, it is usually difficult to train neural networks as well [86].

2.3.5 Frequency warping based methods

In statistical parametric methods, the target speech is generated from model parameters, which are estimated from the feature distributions by data-driven. As the model parameters are estimated to minimize the mean square error or maximizing likelihood on the training data, this inevitably results in the averaging predictive performance. Such statistical average effect will introduce over-smoothing problem in voice conversion [15,20,34]. Moreover, the physical principles of the spectrum are not taken into account.

To respond to such concern, frequency warping methods are proposed [55,87–90]. An illustration of frequency warping is presented in Figure 2.10. During offline training, given a source spectrum $x(f)$ and a target spectrum $y(f)$, the frequency warping techniques find a warping function $W(f)$ that minimizes the spectral distance between

29
\( \mathbf{x}(W(f)) \) and \( \mathbf{y}(f) \). At runtime, the warping function \( W(f) \) is applied to source spectrogram frame by frame to generate the target spectrogram.

![Figure 2.10: Illustration of a frequency warping function to warp the frequency axes.](image)

To implement such a kind of warping functions, a number of methods have been proposed. In [41], a dynamic frequency warping method was proposed, where a dynamic programming technique was employed to find the frequency warping path that achieves the lowest spectral distortion. Different from the nonparametric frequency warping methods, several parametric frequency warping techniques were studied in [87] within the framework of vocal tract length normalization (VTLN). To build a frequency warping function that generates smooth spectral trajectories, in [55], a weighted frequency warping technique was proposed, where a GMM is employed to produce a weighted frequency warping function. Since frequency warping based conversion methods do not remove any detail of the source spectrum, they produce high quality converted speech which sounds natural. However, the basic frequency warping techniques do not manipulate the relative amplitude of the spectrum, the meaningful parts of the spectrum related to speaker individuality hence will be kept unchanged, as a result, speaker identity conversion performance will be affected considerably.

To modify the relative amplitude of the warped spectra, an amplitude scaling technique was proposed in [88] to compensate the amplitude mismatch and to enhance the
Chapter 2. Background

The objective of amplitude scaling is to compensate the difference between the warped source spectra and the reference target spectra, in this way, the warped source spectra are expected to match the spectral shapes as well as the amplitude of the target speech. Amplitude scaling is usually performed on the log-magnitude spectra, as log-magnitude spectra are more perceptually relevant to human hearing, and are more effective in capturing voice timbre [88]. Frequency warping with amplitude scaling was shown to achieve higher converted speech quality than GMM-based methods while achieves similar speaker identity conversion performance to the classic GMM-based methods [88, 89]. Similar observations were reported in [89], where maximum likelihood GMM method with dynamic feature constraint and global variance (GMM-GV) of the target features was used as the reference baseline. It confirms that the frequency warping with amplitude scaling method is able to achieve similar performance as GMM-GV but using fewer number of parameters.

Even though frequency warping based methods can obtain the target speech with very high speech quality, they still have limitations in the identity conversion. Frequency warping does not allow formant splitting or merging which is usually required in spectral mapping [91]. Compensating the amplitude mismatch between warped and target spectra is also a challenge, as the amplitude scaling is affected by the frequency axis alignment performance.

2.3.6 Unit selection based methods

In statistical parametric methods, the spectral parameters are generated from statistical models, while in frequency warping based methods, the frequency axes of the source spectrum are shifted towards that of the target spectrum. Different from statistical parametric and frequency warping methods, unit selection based methods utilise original target speaker's feature vectors directly to generate the target speech.

The work of unit selection methods for voice conversion is inspired by unit selection based speech synthesis, which was proposed in [92] to automatically select and concatenate target speech segments to generate a speech signal. In [51], unit selection method, which uses source speech as reference speech to select the target units, is proposed for text-independent voice conversion. In [93], the authors improved the original unit selection approach [51] by using JD-GMM based converted speech as reference.
speech. To avoid discontinuities at the concatenation boundaries, the unit selection methods [51, 93] consider both the target cost and concatenation cost. Unfortunately, they only use one frame to calculate the concatenation cost, which has not considered a smooth frame-to-frame transition in the target space. In addition, temporal information is also ignored in the generated speech parameter sequence, which will result in the discontinuity of the concatenation points and affect the perceptual quality of the synthesized speech.

A major issue in most of the conventional voice conversion methods is that they assume that the short-term frames are independent observations of each other. Inspired by the findings in exemplar-based speech recognition [94] which considers the dependency of multiple frames, an exemplar-based unit selection method was proposed in [95] to avoid frame-by-frame independence assumption. Exemplars which span over a fixed number of frames are used as the basic units to calculate the concatenation cost and to generate target speech parameters to avoid discontinuity at the concatenation boundaries. The process is summarized as follows: First, source-target exemplar pairs are found on parallel training data. Then, several target candidate exemplars are selected for each source exemplar in a source sentence resulting in an exemplar network, and at the same time, target cost and concatenation cost are calculated within network. After that, Viterbi algorithm is adopted to find the optimal target exemplar sequence which minimizes the overall target and concatenation costs as illustrated in Figure 2.11. Finally, the converted speech parameters are generated from the overlapping exemplars by considering temporal information constraint.

Although unit-selection approaches can generate converted speech with better speaker similarity to the target speaker, a phonetically balanced and phonetically rich dataset is generally required to ensure phonetic coverage. Such a large dataset is expensive to collect in practice.

2.4 Applications

Voice conversion technology has many applications, such as personalized speech synthesis [59], speech enhancement [7, 8], speaking-aid [9, 10] and spoofing attack [6]. This section briefly introduces the applications in text-to-speech synthesis and spoofing attacks which are related to the contributions of this thesis.
Figure 2.11: Illustration of searching for the optimal exemplar sequence. In the figure, dashed line (connecting $X^{(t)}$ and $Y^{(t)}_k$) represents target cost and solid line (connecting $Y^{(t)}_k$ and $Y^{(t+1)}_j$) represents concatenation cost.

2.4.1 Text-to-speech synthesis

The most important application of voice conversion is to personalize a text-to-speech synthesis system [59]. In practice, a large amount of high-quality speech from the specific speaker is required to build a high-quality text-to-speech synthesis system. However, it is expensive to collect a large amount of speech data from every speaker.

To integrate voice conversion with unit-selection based speech synthesis, a voice conversion system can be used as a module to post-process the speech signal from the text-to-speech system to generate a specific speaker’s voice. Figure 2.12 presents an illustration of a way to combine a text-to-speech synthesis system and a voice conversion system. In such an integration, the voice from the speech synthesis system is used as the source training data for voice conversion, while the specific speaker’s voice is used as the target training data. No modification is required for the speech synthesis in order to integrate a voice conversion module.

The integration of voice conversion with statistical parametric speech synthesis is straightforward and is much easier than that with unit-selection speech synthesis, as the voice conversion functions can be applied to the speech parameters from the statistical parametric speech synthesis system directly without extracting feature representation from the speech signals generated from the speech synthesis system.
2.4.2 Spoofing attack

The objective of a speaker verification system is to automatically accept or reject a claimed identity $S$ of one speaker based on just the speech sample $X = \{x_1, x_2, \cdots, x_t, \cdots, x_T\}$ [4]. This verification process is illustrated in Fig. 2.13 and is formulated as a hypothesis test:

$$\Lambda(X) = \frac{p(X|\lambda_H)}{p(X|\lambda_{\tilde{H}})},$$

(2.19)

where $\lambda_H$ is the model parameters of hypothesis $H$ that the speech sample $X$ is from speaker $S$, and $\tilde{H}$ is an alternative hypothesis that the speech sample is not from the claimed identity $S$. The likelihood ratio (or likelihood score) $\Lambda(X)$ is used to decide which hypothesis, $H$ or $\tilde{H}$, is true based on a pre-defined threshold.

In general, speaker verification systems make the binary decision based on the
feature distributions. On the other hand, given a sequence of features $Y$ from an attacker, voice conversion technology can project the attacker’s features to the target speaker’s feature space through the mapping function $\hat{X} = \mathcal{F}(Y)$, and in this way, the speaker verification systems might be deceived by the generated target features $\hat{X}$.

### 2.5 Summary

In this chapter, a general concept of speaker characteristic and voice conversion is first introduced as the fundamental knowledge. Then, each module in a typical voice conversion framework is briefly discussed. After that, a literature review of current state-of-the-art spectral mapping methods is presented with a focus on the Gaussian mixture model based method, as GMM-based method is one of the most popular methods for spectral conversion, and is a well established baseline system for the contributions in Chapters 3 and 4. Finally, two applications of voice conversion, namely text-to-speech synthesis and spoofing attack, are briefly discussed. These information provide a fundamental knowledge of this thesis and background knowledge for the contributions in Chapters 3, 4 and 5.
Chapter 3

Mixture of Factor Analysers Using Priors from Nonparallel Speech

A robust voice conversion function relies on a large amount of parallel training data, which are expensive or not possible to collect in practical applications. This situation is called the data sparseness problem. To tackle this issue, this chapter describes an approach to utilise nonparallel data for conversion. In particular, a mixture of factor analysers method integrates prior knowledge from additional nonparallel speech data to train the conversion function with the need of less parallel data. This work has been published in IEEE Signal Processing Letters [19].

The chapter is organised as follows. Section 3.1 presents the motivation and related work; Section 3.2 introduces the proposed mixture of factor analysers method with integrated priors from nonparallel speech; Section 3.3 presents the experimental setups and results. Section 3.4 summaries the contributions and findings of this study.

3.1 Motivation

As discussed in Chapter 2, the implementation of spectral conversion function is one of the important task in voice conversion. To implement a robust voice conversion function, many statistical methods have been proposed, such as mapping codebooks [74], artificial neural networks [44,61], Gaussian mixture model [34,59,68], and partial least squares regression [12]. Currently, the joint density Gaussian mixture model (JD-
GMM) [34, 59, 68] is one of the most effective approaches. Unfortunately, it requires a relatively large amount of parallel training data to avoid over-fitting [60].

There have been reported work on speech [96] and speaker recognition [97] where researchers leverage on existing speech corpora from non-target speakers as prior knowledge to improve performance. Following the same strategy, eigenvoice-based conversion [98], and tensor representation of speaker space [99] are examples of similar successful attempts for voice conversion. However, these methods nevertheless still require a large amount of parallel data to realise a conversion function, making it difficult to implement in practical situations.

In speaker verification, the joint factor analysis (JFA) method [100] decomposes a supervector into phonetic-dependent, speaker-dependent and channel-dependent components, and each component is then represented by a low-dimensional set of factors. Inspired by such an approach, this thesis proposes that similar decomposition may be useful for voice conversion, where phonetic and speaker specific components of speech spectral vectors can be separated, and then factor analysis is applied to model the speaker specific component. Specifically, the speaker specific component can be represented by a low-dimensional set of latent variables via the factor loadings. In the implementation, a mixture of factor analysers (MFA) [101], which was previously used to refine covariance of JD-GMM in voice conversion [11], is adopted to cover the intended speaker space densely.

The main contribution of this work is a new technique to estimate the phonetic component and factor loadings from non-parallel prior data for voice conversion. Specifically, during the training process, only a low-dimensional set of speaker identity factors and a tied covariance matrix are estimated instead of a full conversion function from the source-target parallel utterances. Even though parallel utterances are still required to estimate the conversion function, the use of prior data reduces the need of parallel training samples as compared to the conventional JD-GMM approach [68]. To the best of our knowledge, this is the first study that integrates prior knowledge from non-parallel data into the training of conversion function.
Chapter 3. Mixture of Factor Analysers Using Priors from Nonparallel Speech

3.2 Mixture of factor analyzers

As discussed in Chapter 2, a large amount of parallel training data are required to estimate the model parameters \( \lambda^{(z)} = \{ \pi_k^{(z)} , \mu_k^{(z)} , \Sigma_k^{(z)} | k = 1, 2, \ldots, K \} \) in the joint-density Gaussian mixture model to achieve reliable performance. To overcome this, this chapter proposes to use nonparallel prior data to estimate some of the speaker-independent parameters that are required in the conversion function in advance. By using these speaker-independent parameters, a reduced amount of parallel data from the source and target speakers can then be employed to estimate the speaker-dependent parameters to realise the conversion function.

The basic idea is to model the speaker-dependent and speaker-independent factors separately. To implement this, a spectral vector is assumed to consist of phonetic and speaker specific components, which are statistically independent, and the speaker specific component can then be represented by a low-dimensional speaker identity vector (SIV) via a low-rank factor loading matrix. A factor analysis model is employed to represent this idea:

\[
o = \mu + Tw + \epsilon,
\]

where \( o \in \mathbb{R}^d \) is an observed \( d \)-dimensional spectral vector, \( \mu \in \mathbb{R}^d \) is the speaker-independent phonetic component, \( Tw \) is the speaker specific component in which \( w \in \mathbb{R}^{m \times 1} \) is the latent SIV, \( T \in \mathbb{R}^{d \times m} \) is the factor loading matrix, and \( \epsilon \) is the noise term.

Factor analysis is a linear single-Gaussian latent variable model [102]. Like other generative models, it also models the distributions of the speech data in the form of Gaussian distributions. In practice, the speech data can not be well represented by a single Gaussian. To better represent the speech data, a mixture of factor analysers (MFA) model that was proposed in [101], is employed. A MFA model is closely related to Gaussian mixture model, and the likelihood function of the nonparallel prior data

\[
O = [O^{(1)}, O^{(2)}, \ldots, O^{(s)}, \ldots, O^{(S)}]^\top
\]
Chapter 3. Mixture of Factor Analysers Using Priors from Nonparallel Speech

for the model \( \lambda^{(MFA)} = \{ \pi_k, \mu_k, T_k, \Sigma_k \mid k = 1, 2, \ldots, K \} \) is:

\[
P(O, w \mid \lambda^{(MFA)}) = P(O \mid w, \lambda^{(MFA)}) P(w)
= \prod_{s=1}^{S} P(O^{(s)} \mid w_s, \lambda^{(MFA)}) P(w_s),
\]

where

\[
O^{(s)} = [o^{(s)}_1, o^{(s)}_2, \ldots, o^{(s)}_{N_s}]
\]

\[
P(O^{(s)} \mid w_s, \lambda^{(MFA)}) = \prod_{n=1}^{N_s} \sum_{k=1}^{K} \pi_k \mathcal{N}(o^{(s)}_n \mid \mu_k + T_k w_s, \Sigma_k).
\]

\[
P(w_s) = \mathcal{N}(0, I)
\]

where \( \mathcal{N} \) represents the Gaussian function, \( S \) represents the number of speakers, and \( N_s \) represents the number of frames from the \( s \)-th speaker, \( \mu_1, \mu_2, \ldots, \mu_K \in \mathcal{R}^d \) represent speaker independent phonetic vectors, \( w_s \in \mathcal{R}^m \) is the SIV of speaker \( s \), \( T_k \in \mathcal{R}^{d \times m} \) is the factor loadings of the \( k \)-th factor analyser component with prior probability \( \pi_k \) and \( \sum_{k=1}^{K} \pi_k = 1 \).

The proposed spectral conversion framework is presented in Fig. 3.1. In the offline training process, nonparallel prior data are first employed to estimate the speaker-independent phonetic component \( \mu_k \) and factor loadings \( T_k \) that will be derived in section 3.2.1 and 3.2.2, respectively. Then \( \mu_k \) and \( T_k \) are adopted to jointly estimate the speaker identity vectors \( w^{(x)}, w^{(y)} \) for source and target in section 3.2.3, and finally derive the conversion function, which is similar to the JD-GMM based conversion function by minimising the mean square error as presented in Chapter 2.

3.2.1 Speaker-independent phonetic vectors estimation

In theory, all the parameters \( \lambda^{(MFA)} \) could be estimated at the same time as in [101]. To benefit from a large speaker independent database [100] and ensure that the phonetic vectors are not affected by the speaker-specific component when estimating the factor loadings, a pre-trained GMM is used to represent the phonetic space. While a Gaussian component may not correspond to a phonetic unit exactly, a mixture of Gaussian components is assumed to be able to cover the whole phonetic space. In this way, the
Chapter 3. Mixture of Factor Analysers Using Priors from Nonparallel Speech

Figure 3.1: Proposed spectral conversion system

The likelihood function for the phonetic GMM $\lambda^{(\text{phonetic})} = \{\pi_k, \mu_k, \Sigma_k | k = 1, 2, \ldots, K\}$ is written as,

$$P(O | \lambda^{(\text{phonetic})}) = \prod_{s=1}^{S} \prod_{n=1}^{N_s} \sum_{k=1}^{K} \pi_k \mathcal{N}(o_n^{(s)} | \mu_k, \Sigma_k),$$

(3.7)

where $\mu_k \in \mathcal{R}^d$ is an estimated phonetic vector, and $\Sigma_k \in \mathcal{R}^{d \times d}$ is the covariance matrix. EM algorithm is used to estimate the parameters $\lambda^{(\text{phonetic})}$. The $\pi_k$, $\mu_k$ and $\Sigma_k$ in (3.5) are replaced by that in (3.7), and $\pi_k$, $\mu_k$ and $\Sigma_k$ are fixed when estimating the factor loading matrices $T_k$. 

40
3.2.2 Speaker-independent factor loadings estimation

Given $\lambda^{(\text{phonetic})}$, we use EM algorithm to estimate the factor loading matrices $T_k$ in (3.3), as there are latent variables $w$. The E-step and M-step are written as follows:

**E-step**

Calculate the occupation probability $\gamma_n^{(s)}(k)$ and the expectation of latent variable $w$:

$$
\gamma_n^{(s)}(k) = \frac{\pi_k P(o_n^{(s)}|w_s, T'_k, \mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l P(o_n^{(s)}|w_s, T'_l, \mu_l, \Sigma_l)} \quad (3.8)
$$

$$
\mathcal{E}[w_s] = F^{-1} \cdot \sum_{n=1}^{N_s} \sum_{k=1}^{K} \gamma_n^{(s)}(k) T'_k \Sigma_k^{-1}(o_n^{(s)} - \mu_k) \quad (3.9)
$$

$$
\mathcal{E}[w_s w_s^\top] = F^{-1} + \mathbb{E}[w_s] \mathbb{E}[w_s]^\top, \quad (3.10)
$$

where $\mathcal{E}[w_s]$ is the expectation of $w_s$, $\mathcal{E}[w_s w_s^\top]$ is the expectation of $w_s w_s^\top$, $F = I + \sum_{n=1}^{N_s} \sum_{k=1}^{K} \gamma_n^{(s)}(k) T'_k \Sigma_k^{-1} T'_k$, $F^{-1}$ and $\Sigma_k^{-1}$ are the inverse of $F$ and $\Sigma_k$, respectively, and $T'_k$ are the factor loading matrices $T_k$ estimated in previous M-step.

**M-step**

Estimate the new factor loading matrices $T_k$:

$$
T_k = \frac{\sum_{s=1}^{S} \sum_{n=1}^{N_s} \gamma_n^{(s)}(k) \cdot \mathbb{E}[w_s](o_n^{(s)} - \mu_k)}{\sum_{s=1}^{S} \sum_{n=1}^{N_s} \gamma_n^{(s)}(k) \mathbb{E}[w_s w_s^\top]} \quad (3.11)
$$

In the implementation, the factor loading matrices $T_k$ are randomly initialised at the beginning. On the voice conversion database, the convergence of the EM algorithm goes well in 10 iterations. Hence, 10 EM iterations are applied to estimate the factoring loading matrices $T_k$.

3.2.3 Voice conversion using mixture of factor analyzers

Given the estimated factor loadings and phonetic vectors from the nonparallel priori data, the converted function can be estimated from the parallel voice conversion data.
\( \mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, \ldots, \mathbf{z}_T] \), where \( \mathbf{z}_t = [\mathbf{x}_t^\top, \mathbf{y}_t^\top] \in \mathbb{R}^{2d} \). The phonetic vectors are concatenated as \( \mathbf{\mu}_k^{(z)} = [\mathbf{\mu}_k^x, \mathbf{\mu}_k^y] \in \mathbb{R}^{2d} \) and the factor loadings are formed as \( \mathbf{A}_k = \begin{bmatrix} \mathbf{T}_k & 0 \\ 0 & \mathbf{T}_k \end{bmatrix} \in \mathbb{R}^{2d \times 2m} \). Note that the two \( \mathbf{\mu}_k \) in \( \mathbf{\mu}_k^{(z)} \) are identical and the two \( \mathbf{T}_k \) in \( \mathbf{A}_k \) are also identical. This concatenation will not change the phonetic mapping when training conversion function. Thus the joint distribution for the parallel data can be written as follows.

\[
P(\mathbf{Z}|\mathbf{w}^{(z)}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{z}|\mathbf{\mu}_k^{(z)} + \mathbf{A}_k \mathbf{w}^{(z)}, \Sigma^{(z)}). 
\]

Here \( \mathbf{w}^{(z)} = [\mathbf{w}^{(x)} \top, \mathbf{w}^{(y)} \top] \in \mathbb{R}^{2m \times 1} \) is the joint speaker identity vector where \( \mathbf{w}^{(x)} \in \mathbb{R}^{m \times 1} \) is for source speaker and \( \mathbf{w}^{(y)} \in \mathbb{R}^{m \times 1} \) is for target speaker, and \( \Sigma^{(z)} = \begin{bmatrix} \Sigma^{(xx)} & \Sigma^{(xy)} \\ \Sigma^{(yx)} & \Sigma^{(yy)} \end{bmatrix} \in \mathbb{R}^{2d \times 2d} \) is a covariance matrix. A full covariance matrix consists of a large number of free parameters which need to be estimated. To circumvent the data sparseness issue and avoid numerical problem, a tied covariance matrix shared by all the Gaussians is used in implementation. This method is hence dubbed as \textit{tied mixture of factor analyzers} (TMFA). The benefit of using factor analysis is that the estimation of the speaker specific component contains a low-dimensional parameters, which is relatively small, and hence the SIV requires less training data, since the factor loadings are estimated in advance. Similar as that for equation (3.3), EM algorithm is adopted to estimate \( \mathbf{w}^{(z)} \) and \( \Sigma^{(z)} \) under the maximum likelihood criterion by:

**E-step**

Calculate the occupation probability \( p_k(\mathbf{z}_t) \) and joint speaker identity vector \( \mathbf{w}^{(z)} \):

\[
p_k(\mathbf{z}_t) = \frac{\pi_k P(\mathbf{z}_t|\mathbf{w}^{(z)}, \mathbf{A}_k, \mathbf{\mu}_k^{(z)}, \Sigma^{(z)})}{\sum_{l=1}^{K} \pi_l P(\mathbf{z}_t|\mathbf{w}^{(z)}, \mathbf{A}_l, \mathbf{\mu}_l^{(z)}, \Sigma^{(z)})} \quad (3.13)
\]

\[
\mathbf{w}^{(z)} = \frac{\sum_{t=1}^{T} \sum_{k=1}^{K} p_k(\mathbf{z}_t) \mathbf{A}_k^\top \Sigma^{(z)-1}(\mathbf{z}_t - \mathbf{\mu}_k^{(z)})}{\mathbf{I} + \sum_{t=1}^{T} \sum_{k=1}^{K} p_k(\mathbf{z}_t) \mathbf{A}_k^\top \Sigma^{(z)-1} \mathbf{A}_k^\top} \quad (3.14)
\]
M-step

Estimate the new tied covariance matrix $\Sigma^{(z)}$:

$$\Sigma^{(z)} = \frac{\sum_{t=1}^{T} \sum_{k=1}^{K} p_k(z_t) vv^T}{\sum_{t=1}^{T} \sum_{k=1}^{K} p_k(z_t)},$$

(3.15)

where $v = z_t - \mu_k^{(z)} - A_k w^{(z)}$. In this EM algorithm, the tied covariance matrix is initialised with global covariance matrix and $w^{(z)}$ is initialised to the zero vector. Above EM algorithm is repeated for three iterations to estimate $\Sigma^{(z)}$ and $w^{(z)}$.

In the conversion process, given $x$, the tied joint-density MFA model is adopted to predict the target feature vector $\hat{y} = F(x)$ as follows:

$$F(x) = \sum_{k=1}^{K} p_k(x)\left(\mu_k + T_k w^{(y)} + \Sigma^{(yx)}(\Sigma^{(xx)})^{-1}(x - \mu_k - T_k w^{(x)})\right)$$

where $p_k(x)$ is the occupation probability of $x$ belonging to the $k$-th factor analyser.

3.3 Experiments

To assess the performance of the proposed method, experiments were conducted on the public available database, namely CMU ARCTIC corpus\(^1\). Two speaker pairs, male-to-male (M-M, BDL-to-RMS) and female-to-female (F-F, SLT-to-CLB), were involved in the experiments. In the experiments, the training data from each speaker were varied from 2 to 8 utterances, and the testing data involved 50 utterances from each speaker. The Aurora 4 corpus, which has 83 speakers and each speaker has around 100 utterances (clean speech), was used as the prior data to estimate speaker-independent phonetic vectors and factor loadings.

The speech signal was sampled at 16 kHz. Spectral envelope and fundamental frequency (F0) were extracted by the STRAIGHT system [37] at 5ms step, and the spectral envelope was parameterised as 25-order mel-cepstral coefficients (MCC), including the energy coefficient, which was not converted. Hence only 24-order coefficients were converted by the conversion function. F0 was modified by equalizing the mean and

\(^1\)http://www.festvox.org/cmu_arctic/index.html
Chapter 3. Mixture of Factor Analysers Using Priors from Nonparallel Speech

In this work, the following conversion methods were compared:

1. **GMM-full**: The JD-GMM based conversion method using full covariance matrices in the Gaussian components.

2. **GMM-cross**: The JD-GMM based conversion method with cross-diagonal covariance matrices which have only diagonal and cross-covariance elements that are non-zero [11] as illustrated in Figure 2.8.

3. **TMFA-full**: The proposed TMFA method using full covariance matrices.

4. **TMFA-cross**: The proposed TMFA method using cross-diagonal covariance matrices.

Note that in the experiments, the GMM-full method is comparable with the TMFA-full method in terms of number of parameters, and similarly GMM-cross is comparable with TMFA-cross.

### 3.3.1 Objective evaluation

To evaluate the conversion methods objectively, the mel-cepstral distortion (MCD) was used as the objective evaluation measure. The MCD was calculated between a generated target frame and a reference target frame (ground truth) [34] as follows:

$$
\text{MCD}[\text{dB}] = \frac{10}{\log 10} \sqrt{2 \sum_{d=1}^{24} (c_d - c_d^{(converted)})^2},
$$

where $c_d$ and $c_d^{(converted)}$ are the $d$-th reference target and converted MCCs, respectively. A lower MCD value indicates smaller distortion.

The proposed method was first compared with the JD-GMM method using only two training utterances. The average MCD values of M-M and F-F spectral conversions as a function of the number of factors in TMFA are presented in Fig. 3.2. Due to the limitation of the training data, only one Gaussian component was adopted in the baseline JD-GMM model, as it gave a lower MCD value than that with 2 or 4 Gaussian components. In the TMFA method, there were 128 Gaussian components in the TMFA.
method. With the fixed number of Gaussian components, the number of factors in the factor loadings was varied from 8 to 64 to assess the performance. It is observed that, when more than 24 factors are employed, TMFA gives much lower spectral distortion than the baseline JD-GMM does. Another observation is that TMFA-cross as well as GMM-cross always outperforms TMFA-full and GMM-full that suggests the full covariance models suffer from over fitting. Hence, in the following experiments, only TMFA-cross and GMM-cross are examined and compared.

![Figure 3.2: Average mel-cepstral distortion as a function of the number of factors.](image)

TMFA-cross and GMM-cross are further trained with varying the amount of parallel training data. Here, the number of factors is set to be $m = 44$, which is the median number between 24 and 64. The average MCD values of M-M and F-F conversion are presented in Fig. 3.3. It is observed that if there are only a limited amount of parallel training available, the TMFA method outperforms the JD-GMM method in terms of spectral distortion. This is because when the parallel training data are limited, the TMFA model can benefit from the prior knowledge that is learnt from nonparallel prior
data, while JD-GMM cannot. However, one the limitation of the TMFA method is that when the amount of parallel training data increases, the TMFA model does not work as good as JD-GMM. This is because the TMFA model has few parameters and hence has less freedom to fit the training well.

![Figure 3.3: Average MCD in terms of number of training utterances](image)

### 3.3.2 Subjective evaluation

Subjective listening tests were also conducted to assess the performance of the proposed method. In the subjective listening test, TMFA-cross with 44 factors was compared with GMM-cross with 1 mixture, and the number of training utterances was set to be two.

An AB preference test was first conducted to assess the similarity of the converted speech. In this test, two converted speech samples (A and B) from the two conversion methods were randomly presented to the subjects and then the reference target speech
Chapter 3. Mixture of Factor Analysers Using Priors from Nonparallel Speech

was presented to the subjects, after that the subjects were asked to choose the one, A or B, which sounded more similar to the reference target speech by by choosing one of the followings: 1) A is more similar; 2) B is more similar; 3) no preference.

In the test, 10 sentences randomly selected from the evaluation set were used, and 8 subjects participated in the listening test. The similarity preference results with 95% confidence intervals are shown in Fig. 3.4. It is observed that the proposed TMFA method outperforms the JD-GMM method significantly in terms of speaker similarity. It confirms the effectiveness of the proposed method.

Figure 3.4: AB preference test results for speaker similarity with 95% confidence intervals.

An AB preference test was also conducted to evaluate the perceptual quality of the converted speech samples. The evaluation process is similar to the similarity test. However, there were only two choices: 1) Sample A is much better; 2) Sample B is much better. Same as the similarity test, 8 subjects listened to the 10 sentences pairs. The quality preference test results with 95% confidence intervals are presented in Fig. 3.5. It shows that the proposed TMFA method outperforms the JD-GMM method significantly in terms of perceptual speech quality.
Chapter 3. Mixture of Factor Analysers Using Priors from Nonparallel Speech

Figure 3.5: AB preference test results for speech quality with 95% confidence intervals.

3.4 Conclusion

In this chapter, a voice conversion technique based on mixture of factor analysers was proposed to utilise priors learned from the nonparallel training data. The proposed method assumes that a speech spectral vector consists of speaker-independent phonetic and speaker-dependent components. As a result, nonparallel training data can be used to estimate the speaker-independent components to cover the speaker-independent phonetic space. The objective spectral distortion and subjective listening evaluations confirmed the effectiveness of the proposed TMFA method over the conventional JD-GMM method.

However, there are some limitations in the proposed TMFA method. First, parallel training data are still used in the proposed method. It would be interesting to investigate a new way to relax this constraint, and use nonparallel data for training. Second, when sufficient training data are available, the proposed TMFA method is not as good as the JD-GMM method, as the proposed method shares only one speaker-dependent vector which contains limited parameters. The future work will include the tying across factor analysers.
Chapter 4

Exemplar-based sparse representation with residual compensation for voice conversion

This chapter proposes a nonparametric framework for voice conversion, that is, exemplar-based sparse representation with residual compensation. In this framework, a spectrogram is reconstructed as a weighted linear combination of speech segments, called exemplars, which span multiple consecutive frames. The linear combination weights are constrained to be sparse to avoid over-smoothing, and high-resolution spectra are employed in the exemplars directly without dimensionality reduction to maintain spectral details. In addition, a spectral compression factor and a residual compensation technique are included in the framework to enhance the conversion performances. This work has been published in the 8th ISCA Speech Synthesis Workshop (SSW8) [20], and its extended version is published in the IEEE/ACM Transactions on Audio, Speech and Language Processing [21].

The chapter is organized as follows: Section 4.1 provides the motivation of this work; Section 4.2 introduces the proposed exemplar-based sparse representation with residual compensation method; Section 4.3 presents the experimental setups and results; Section 4.4 discusses the relationships of the proposed method with the conventional voice conversion methods; Section 4.5 summaries the contributions of this chapter and discuss the possible future work to extend the proposed method.
4.1 Motivation

A large number of statistical parametric and frequency warping approaches have attempted to achieve a robust spectral mapping [8, 12, 34, 44, 45, 55, 59, 61, 68, 87–89, 103] for voice conversion applications. However, the robustness of statistical parametric approaches is limited by the fact that they attempt to predict speech trajectories from model parameters. Robustness refers to the ability that a system is capable to handle new training data without much tuning. In the parametric approaches, when new training training data arrives, the model parameters for prediction are required to be re-optimized to reduce the mismatch between the ‘old’ model and new data. Robustness also refers to the ability that a method is reliable to handle various training scenarios, such as limited training samples and high-dimensional features. When there are too many parameters and too few training samples, over-fitting occurs [12]. Over-fitted model usually has poor predictive performance.

In addition to the issue of robustness, in conventional statistical parametric approaches, low-resolution features such as mel-cepstral coefficients (MCCs) [104] and line spectrum pair (LSP) [58] are commonly used to represent high-resolution spectra, which are the spectra/spectral envelopes extracted from the discrete Fourier transform or linear predictive coding. The use of low-resolution features is to reduce the feature dimensionality for computational efficiency and robust modeling. However, the use of such low-resolution features loses spectral details and results in converted spectrum that is smoothed. Fig. 4.1 shows a comparison of an original spectral envelope and a reconstructed spectral envelope from 24-order MCCs. It is observed that after reconstruction from low-resolution features, the spectral details are lost, especially in high-frequency bands. There are partial evidences showing that high-resolution feature representations produce synthetic speech with better quality than low-resolution features [17, 18]. In this study, exemplar-based sparse representation with residual compensation is proposed as an alternative nonparametric framework for voice conversion. In this framework, each speech segment is reconstructed as a weighted linear combination of a set of basis exemplars. An exemplar is defined as a speech segment spanning multiple frames extracted from training data, and the set of linear combination weights compose an activation vector, which is constrained to be sparse. There are three advantages of using speech segments in a dictionary: a) it is easy to construct
Chapter 4. Exemplar-based sparse representation with residual compensation for voice conversion

Figure 4.1: Illustration of the smoothing effect of low-resolution features. The dashed line is reconstructed from 24-order MCCs, which are computed from the solid line.

the dictionary by extracting speech segments from the training data; b) it allows us to model high-dimensional spectra directly to maintain the spectral details; and c) the generation of converted spectrogram is straightforward, by simply combining a set of basis speech segments without mapping or modification.

In the proposed exemplar-based sparse representation framework, it is assumed that a collection of acoustically aligned source and target exemplars, called coupled dictionary, could share the same activation vector to generate the converted spectrograms. Due to the nonnegative nature of a spectrogram, in practice, nonnegative matrix factorization (NMF) [105] with sparsity constraint technique [20, 106, 107] has been employed to estimate the activations. The advantage of constraining the activation vector to be sparse helps to avoid the over-smoothing problem that may occur during the linear combination of exemplars.

Within the framework, two techniques, namely spectral compression and residual compensation, are adopted to enhance the converted speech quality. First, inspired by the work in source separation [108], the dynamic range of the spectrogram amplitude affects the performance considerably, as in the original spectrogram, high-frequency bands have low-intensity observations, while low-frequency bands have high-intensity observations. As a result, some important but low-intensity observations will be overlooked. To this end, a spectral compression factor is introduced to balance the intensity
between high- and low-frequency bands.

Second, during the estimation of activations there is inevitably some modelling error between the source spectrogram and the modeled spectrogram. These residuals usually contain some spectral details which are possible to affect the converted speech quality. This thesis hence proposes a residual compensation technique to reimburse the source model residual for the converted spectrogram to enhance the speech quality.

The main contributions of this work are threefold:

• An exemplar-based sparse representation framework for voice conversion is proposed, allowing us to model high-resolution spectra directly.

• A spectral compression method is investigated to emphasize the important but low-intensity high frequency observations.

• A residual compensation technique is introduced to enhance the converted speech quality.

4.2 Exemplar-based sparse representation with residual compensation

To overcome the limitations of statistical parametric approaches, we propose an alternative nonparametric framework, specifically, an exemplar-based sparse representation with residual compensation method, where high-resolution spectra are directly used to synthesize the converted speech. The proposed framework is described in this section.

4.2.1 Basic exemplar-based sparse representation

The idea of exemplar-based sparse representation is to describe a magnitude spectrum as a linear combination of a set of basis spectra, namely, exemplars. Mathematically, it is written as

\[
x^{(\text{DFT})} \approx \sum_{n=1}^{N} a_n^{(\text{DFT})} \cdot h_n = A^{(\text{DFT})} h \quad \text{subject to} \quad h \geq 0,
\]  

(4.1)
where $\mathbf{x}^{(\text{DFT})} \in \mathcal{R}^{F \times 1}$ represents the high-resolution spectrum of one speech frame, $F$ is the dimensionality of high-resolution spectra, $N$ is the number of exemplars in a dictionary, $\mathbf{A}^{(\text{DFT})} = [\mathbf{a}_1^{(\text{DFT})}, \mathbf{a}_2^{(\text{DFT})}, \ldots, \mathbf{a}_N^{(\text{DFT})}] \in \mathcal{R}^{F \times N}$ is the dictionary of exemplars built from the source training set, $\mathbf{a}_n^{(\text{DFT})}$ is the $n$th source exemplar which has the same dimensionality as $\mathbf{x}^{(\text{DFT})}$, $\mathbf{h} = [h_1, h_2, \ldots, h_N] \in \mathcal{R}^{N \times 1}$ is the activation vector and $h_n$ is the nonnegative weight, or activation, of the $n$th exemplar.

Each observation is modeled independently, and a spectrogram of each source utterance can therefore be represented as

$$\mathbf{X}^{(\text{DFT})} \approx \mathbf{A}^{(\text{DFT})} \mathbf{H}, \quad (4.2)$$

where $\mathbf{X}^{(\text{DFT})} \in \mathcal{R}^{F \times M}$ is the source spectrogram, $M$ is the number of frames in a source utterance and $\mathbf{H} \in \mathcal{R}^{N \times M}$ is the activation matrix, each column vector of which is an activation vector in Eq. (4.1).

To generate a converted spectrogram, we assume that paired source-target dictionaries $\mathbf{A}^{(\text{DFT})}$ and $\mathbf{B}^{(\text{DFT})}$ with acoustically aligned exemplars can share the same activation matrix $\mathbf{H}$. Note that each column vector in $\mathbf{B}^{(\text{DFT})}$ corresponds to a column vector in $\mathbf{A}^{(\text{DFT})}$, and they are obtained from the aligned data. Thus, the converted spectrogram can be generated as

$$\hat{\mathbf{Y}}^{(\text{DFT})} = \mathbf{B}^{(\text{DFT})} \mathbf{H}, \quad (4.3)$$

where $\hat{\mathbf{Y}}^{(\text{DFT})} \in \mathcal{R}^{F \times M}$ is the converted spectrogram, $\mathbf{B}^{(\text{DFT})} \in \mathcal{R}^{F \times N}$ is the target dictionary of exemplars from target training data, and $\mathbf{H}$ is as found in Eq. (4.2).

Due to the nonnegative nature of source spectrogram $\mathbf{X}^{(\text{DFT})}$ and source dictionary $\mathbf{A}^{(\text{DFT})}$, the nonnegative matrix factorization (NMF) technique [105,106] is employed to estimate the activation matrix $\mathbf{H}$, which is found by minimizing the objective function

$$\mathbf{H} = \arg \min_{\mathbf{H} \geq 0} d(\mathbf{X}^{(\text{DFT})}, \mathbf{A}^{(\text{DFT})} \mathbf{H}) + \lambda \| \mathbf{H} \|_1, \quad (4.4)$$

where $\lambda$ is the sparsity penalty factor. In practice, the generalised Kullback-Leibler (KL) divergence is used for $d(\mathbf{X}^{(\text{DFT})}, \mathbf{A}^{(\text{DFT})} \mathbf{H})$. Similar to [106], we minimize the objective function in Eq. (4.4) by iteratively applying the following multiplicative
Chapter 4. Exemplar-based sparse representation with residual compensation for voice conversion

updating rule:

\[ H \leftarrow H \odot \frac{A^{(DFT)^\top} X^{(DFT)}}{A^{(DFT)^\top} + \lambda}, \]  

(4.5)

where \( \odot \) represents element-wise multiplication and divisions are also element-wise. The convergence of Eq. (4.4) using this update rule is proven in [106]. In our study with real speech data, it is observed that this update rule converges robustly.

In the following, this thesis makes several modifications to the basic setup described above.

4.2.2 Spectrum compression

The relative range between high- and low-intensity observations is an important factor which affects the activation matrix estimation, as well as spectrogram generation. In the context of source separation, changing the dynamic range of spectrograms by exponentiating them has been found to affect the performance significantly [108]. In a similar way, we introduce a spectral compression parameter \( \rho \) into the computation of the activation matrix, as follows:

\[ (X^{(DFT)})^\rho \approx (A^{(DFT)})^\rho H, \]  

(4.6)

\[ \hat{Y}^{(DFT)} = ((B^{(DFT)})^\rho H)^{1/\rho}. \]  

(4.7)

An analysis of the spectral shapes was conducted by varying the compression factor from 0.2 to 1.0 as presented in Fig. 4.2. It shows that a smaller compression factor implies more emphasis on higher frequency bands that have low intensity. Note that the spectral compression will not change the nonnegative nature of a spectrogram.

Values \( \rho < 1 \) compress the dynamic range of a spectrum, making its values closer to each other. The KL divergence is linear in terms of the scale of its arguments [109], and compression/expansion affects the relative weight of small/large intensity observations in the estimation. Note that \( \rho \) can be applied to the source spectrogram and dictionaries in advance before estimating the activation matrix. This allows us to use the same objective function and updating rule as in Eqs. (4.4) and (4.5), to find the activation matrix \( H \). When \( \rho = 1 \), Eqs. (4.6) and (4.7) are reduced to Eqs. (4.2) and (4.3), respectively.

54
4.2.3 Contextual information

So far, as shown in Eqs. (4.2) and (4.6), no contextual information is taken into consideration when estimating the activation matrix, in other words, each frame is modeled independently. It is easy to understand that contextual information is important in modeling a speech signal. To benefit from the context, we suggest using a speech segment that spans multiple consecutive frames as an exemplar in the source dictionary. The column vectors in each exemplar are stacked into a single vector to simplify the notation. In this way, an exemplar from the dictionary in Eqs. (4.1), (4.2) and (4.6) can be defined as

\[
a^{(\text{DFT})}_n = [a^{(\text{DFT})}_{n,q}; a^{(\text{DFT})}_{n,-1}; a^{(\text{DFT})}_{n,0}; a^{(\text{DFT})}_{n,+1}; \ldots; a^{(\text{DFT})}_{n,q}]
\]

where \( L = 2 \times q + 1 \) is the window size of an exemplar, \( a^{(\text{DFT})}_{n,0} \) is exactly the same as \( a^{(\text{DFT})}_n \) in Eq. (4.2), \( a^{(\text{DFT})}_{n,-q} \) and \( a^{(\text{DFT})}_{n,+q} \) are the \( q \)th frames preceding and following frame \( a^{(\text{DFT})}_{n,0} \) in the original time sequence, respectively. Thus, the stacking vector \( a^{(\text{DFT})}_n \in \mathbb{R}^{(L \times F) \times 1} \) is able to represent an exemplar spanning \( L \) frames.
4.2.4 Using low-resolution features for faster computation

As shown in Eqs. (4.2), (4.3), (4.6) and (4.7), the size of the activation matrix $\mathbf{H}$ is independent of the feature dimensionality of the source dictionary $\mathbf{A}^{(\text{DFT})}$. On the other hand, the feature dimensionality of $\mathbf{A}^{(\text{DFT})}$ will affect the computation and memory usage considerably, especially when a relatively large context is used. To overcome this, we propose a new implementation of NMF using low-resolution features in the source dictionary. This kind of coupled dictionaries has previously been applied e.g. to combine good time and frequencies resolutions [110], to expand the bandwidth of speech [111] and to do robust automatic speech recognition [106].

Let us consider an exemplar without contextual information first and define the low-resolution implementation as

$$
\mathbf{W}(X^{(\text{DFT})})^\rho \approx \mathbf{W}(A^{(\text{DFT})})^\rho \mathbf{H},
$$

(4.9)

where $\mathbf{W} \in \mathcal{R}^{U \times F}$ is the matrix to perform dimensionality reduction, and $U$ is the dimensionality of the low-resolution feature with $U \leq F$ constraint.

In practice, the low-resolution used here corresponds to the mel-scale, so that each column of $\mathbf{W}$ is the triangular magnitude response of a filter, and thus, for simplicity, we use $X^{(\text{MEL})} \in \mathcal{R}^{U \times M}$ for $\mathbf{W}(X^{(\text{DFT})})^\rho$, and denote $\mathbf{W}(A^{(\text{DFT})})^\rho = \mathbf{A}^{(\text{MEL})} \in \mathcal{R}^{U \times N}$. In this way, Eq. (4.9) becomes:

$$
X^{(\text{MEL})} \approx \mathbf{A}^{(\text{MEL})} \mathbf{H}.
$$

(4.10)

Note that Eq. (4.10) is similar to Eq. (4.2), this allows us to use the same estimation method. To benefit from multiple-frame exemplars, we follow the same stacking as in Eq. (4.8) to establish a context-dependent source dictionary by simply replacing $(a_n^{(\text{DFT})})^\rho$ with $a_n^{(\text{MEL})} = \mathbf{W}(a_n^{(\text{DFT})})^\rho$.

The advantage of using low-resolution features rather than high-resolution features to estimate the activation matrix is that the computational complexity can be reduced greatly. Even though low-resolution features are used to estimate the activation matrix as Eq. (4.10), the activations are applied on the high-resolution dictionary $\mathbf{B}^{(\text{DFT})}$ to generate the converted spectrogram as Eq. (4.7). Note that the low-resolution source dictionary $\mathbf{A}^{(\text{MEL})}$ is acoustically aligned with the high-resolution dictionary $\mathbf{B}^{(\text{DFT})}$,
we hence assume that $B^{(\text{DFT})}$ can also share the activation matrix $H$ with $A^{(\text{MEL})}$.

Figure 4.3: Illustration of the exemplar-based sparse representation with residual compensation framework, which consists of four processes: (1) estimating the activation matrix $H$ using low-resolution features; (2) calculating the source residuals $R^{(X)}$ following Eq. (4.11); (3) mapping source residuals to target residuals following Eq. (4.13); and (4) generating converted spectrograms $\hat{Y}^{(\text{DFT})}$ following Eq. (4.14). In the diagram, $U$ and $F$ are the feature dimensionalities of low-resolution and high-resolution features, respectively, $3 \times U$ means an exemplar spanning three frames, $M$ is the number of frames in a source utterance, and $N$ is the number of exemplars in a dictionary. As discussed in Section 4.2.6, $U$ is set to 50, and $F$ is set to 513.

### 4.2.5 Compensating model residual

There is inevitably some modeling error, also called residual, between the observed source spectrogram $X^{(\text{DFT})}$ and the modeled spectrogram $((A^{(\text{DFT})})^{\rho}H)^{1/\rho}$. We hence propose a residual compensation technique to enhance the spectral mapping performance.

To perform residual compensation, during offline training, we use a development set in the following four steps to calculate source and target spectrogram residuals:

(a) Estimate the activation matrix $H$ using low-resolution features as that in Eq. (4.10).

(b) Apply $H$ to the source and target spectral dictionaries to reconstruct source and target spectrograms, respectively.
Chapter 4. Exemplar-based sparse representation with residual compensation for voice conversion

(c) Calculate source residuals $R^{(X)}$ by subtracting the magnitude of the modeled spectrograms with the corresponding reference source spectrograms as

$$R^{(X)} = \log(X^{(\text{DFT})}) - \log(((A^{(\text{DFT})})^\rho H)^{1/\rho}). \quad (4.11)$$

(d) Calculate target residuals $R^{(Y)}$ by subtracting the magnitude of the converted spectrograms with the corresponding reference target spectrograms as

$$R^{(Y)} = \log(Y^{(\text{DFT})}) - \log(((B^{(\text{DFT})})^\rho H)^{1/\rho}). \quad (4.12)$$

The source-target residual pairs can be obtained by using the corresponding reference source-target frame alignment information. With the paired source-target residuals, a mapping can be established as

$$R^{(Y)} \approx F(R^{(X)}). \quad (4.13)$$

In practice, the mapping is implemented by partial least square (PLS) regression, which is able to handle high-resolution features. The details of PLS regression can be found in [12].

The runtime conversion process is illustrated in Fig. 4.3. Similar to that in training, the activation matrix $H$ is first estimated using low-resolution features as that in Eq. (4.10). Next, $H$ is applied to the source and target spectral dictionaries to generate reconstructed source and converted spectrograms, respectively. Then, source residuals are calculated by subtracting the modeled spectrograms with reference source spectrograms in log-scale as that in Eq. (4.11). After that, the source residuals are mapped to target, which are added to the converted spectrograms on a logarithmic amplitude scale. Finally, the residual compensated spectrograms will be reverted back to a linear amplitude scale. The whole process is formulated as

$$\hat{Y}^{(\text{DFT})} = \exp(\log(((B^{(\text{DFT})})^\rho H)^{1/\rho}) + F(R^{(X)})). \quad (4.14)$$

Residual compensation is performed on the logarithmic amplitude scale to guarantee the nonnegative nature of a spectrogram.
4.2.6 Dictionary Construction

As shown in Fig. 4.3, two kinds of dictionaries are involved in this work. Before constructing dictionaries, the feature representations used in this work are presented as follows:

- **High-resolution Magnitude spectra** consist of a sequence of 513-dimensional spectral envelopes extracted by STRAIGHT [37], and the envelopes are passed to STRAIGHT to reconstruct an audible speech signal at runtime. The frequency resolution of the high-resolution spectra is similar to that from the discrete Fourier transform (DFT). To this end, we use label DFT to denote the high-resolution spectra used in the source spectrogram $\mathbf{X}^{(\text{DFT})}$, the target spectrogram $\mathbf{Y}^{(\text{DFT})}$ or $\hat{\mathbf{Y}}^{(\text{DFT})}$, the source spectral dictionary $\mathbf{A}^{(\text{DFT})}$ and the target spectral dictionary $\mathbf{B}^{(\text{DFT})}$.

- **Low-resolution Mel-scale filter-bank energies (MELs)** are obtained by passing the high-resolution magnitude spectrogram to 50 Mel-scale filter-banks, where the lower and upper frequencies are set to be 133.33 Hz and 6,855.5 Hz, respectively. In this chapter, MELs are used as low-resolution features in $\mathbf{X}^{(\text{MEL})}$ and in the source feature dictionary $\mathbf{A}^{(\text{MEL})}$, which are used to estimate the activation matrix.

- **Mel-cepstral coefficients** (MCCs) are obtained by applying the Mel-cepstral analysis technique [112] on a magnitude spectrogram and keeping 24 dimensions as features. During synthesis, MCCs are reverted back to a magnitude spectrogram, which is then passed to STRAIGHT to reconstruct an audible speech signal.

Given a parallel dataset between source and target speakers, we take the following steps to extract paired exemplars:

(a) Extract high-resolution magnitude spectrograms (spectral envelopes) from both source and target speech signals using STRAIGHT.

(b) Apply Mel-cepstral analysis technique [112] to the magnitude spectrogram to compute MCCs.

(c) Apply 50 Mel-scale filter-banks to the source spectrograms to compute 50-dimensional MELs.
(d) Perform dynamic time warping (DTW) to align the source and target MCCs to obtain frame-by-frame source-target alignment.

(e) Apply the frame alignment information to the source and target magnitude spectrograms to obtain the high-resolution source $\mathbf{a}_n^{(\text{DFT})}$ and target exemplars $\mathbf{b}_n^{(\text{DFT})}$, respectively.

(f) Apply the same frame alignment information to the source MELs to obtain the low-resolution source dictionaries $\mathbf{a}_n^{(\text{MEL})}$.

In the above six steps, we produce the paired exemplars from the training dataset. A simple way to construct dictionaries is putting all these paired exemplars as column vectors in the corresponding dictionaries such as $\mathbf{A}^{(\text{DFT})}$, $\mathbf{B}^{(\text{DFT})}$ and $\mathbf{A}^{(\text{MEL})}$. In the experiments, we examine the performance of dictionaries using a subset of the paired exemplars, for example, randomly selecting exemplar pairs and storing as column vectors in corresponding dictionaries.

4.3 Experiments

Experiments were conducted using the VOICES\(^1\) database [113] to assess the performance of the proposed exemplar-based sparse representation with residual compensation method. Speech data from two male speakers (jal and jcs) and two female speakers (sas and leb) was used. Voice conversion was conducted for all the 12 speaker pairs including 4 intra-gender and 8 inter-gender conversions. In each pair, 10 utterances were randomly selected as a training set, 10 utterances as a development set, and 20 utterances as an evaluation set. There was no overlapping across the three sets.

4.3.1 Reference methods and setups

To validate the proposed approaches, a large set of state-of-the-art methods were used as the reference baselines, that include the well established ML-GMM method and several variations of the partial least squares (PLS) regression based methods. Note that PLS-based methods also depend on GMM in order to implement local transformations, but use more advance techniques to obtain the mapping function. In addition,
we implement several nonnegative matrix factorization (NMF) based methods within
the exemplar-based sparse representation framework to show the intermediate methods
towards the proposed method. They are summarized as follows:

- **ML-GMM**: The joint-density Gaussian mixture model (JD-GMM) method with
dynamic feature constraint proposed by Toda et al. [34] is a well-established
baseline method. Note that in this work, 24-dimensional MCCs were used to
represent the spectral envelope. Cross-diagonal covariance was adopted in the
JD-GMM.

- **ML-GMM-GV**: The ML-GMM method with global variance enhancement in [34].
The same configuration as ML-GMM was used and a postprocessing technique
was employed to perform the GV implementation as presented in [114].

- **DKPLS**: The dynamic kernel partial least squares (DKPLS) regression method
has been shown to be effective for implementing a nonlinear conversion function
[45]. The same configuration as that in [45] was used.

- **DKPLS-DFT**: The dynamic kernel partial least squares (DKPLS) regression
method was also applied to high-resolution spectrograms. Comparing with DKPLS,
this implementation is to examine the flexibility of DKPLS in face of high-
dimensional features.

- **DPLS-DFT**: The partial least squares (PLS) regression method [12] was applied
to high-resolution spectrograms, and three consecutive spectra were stacked as
source features to include dynamic information for predicting a target spectrum.
Comparing with DKPLS-DFT, this method is to evaluate the performance of
basic PLS without kernel transformation.

- **NMF-DFT**: This is the basic exemplar-based sparse representation method im-
plemented by nonnegative matrix factorization (NMF). High-resolution magni-
tude spectra were employed in the source and target dictionaries. It used Eq.
(4.2) to estimate the activation matrix and Eq. (4.3) to generate the converted
spectrograms.
• **NMF-DFT-SC**: This is the NMF-DFT method with spectral compression, as presented in Eq. (4.6) and (4.7). With reference to NMF-DFT, this method is to show the effect of spectral compression.

• **NMF-MEL-SC**: This is the NMF with spectral compression method using low-resolution Mel-scale filter-bank energies (MELs) in the source dictionary. It employed Eq. (4.10) to estimate the activation matrix, and Eq. (4.7) to produce the converted spectrograms. With reference to NMF-DFT-SC, this method is to show the effect of feature dimensionality reduction in the source dictionary.

• **NMF-MEL-SC-RC (Proposed)**: This is the complete exemplar-based sparse representation with residual compensation method as presented in Section 4.2.5 and Fig. 4.3. With reference to NMF-MEL-SC, the effect of residual compensation is shown in this setup.

Table 4.1 summarizes the voice conversion methods with involved feature representations, and the equations for activation matrix estimation and spectrogram generation. The spectral mapping was performed using above methods, while F0 was converted by a simple linear conversion, normalizing the mean and variance of the source speech to equalize that of the target. This work only deals with the magnitude spectra, while adopting minimum-phase for all the methods when reconstructing the speech signals. In practice, the STRAIGHT vocoder was employed. For a fair comparison, all the methods shared the same frame alignment obtained from frame-by-frame DTW.

Table 4.1: Summary of the implemented methods and their formulations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Spectral feature</th>
<th>Activation estimation</th>
<th>Spectrogram generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-GMM</td>
<td>MCCs</td>
<td>n/a</td>
<td>Eq. (39) in [34]</td>
</tr>
<tr>
<td>ML-GMM-GV</td>
<td>MCCs</td>
<td>n/a</td>
<td>Eq. (16) in [114]</td>
</tr>
<tr>
<td>DKPLS</td>
<td>MCCs</td>
<td>n/a</td>
<td>Eq. (6) in [45]</td>
</tr>
<tr>
<td>DKPLS-DFT</td>
<td>DFTs</td>
<td>n/a</td>
<td>Eq. (6) in [45]</td>
</tr>
<tr>
<td>DPLS-DFT</td>
<td>DFTs</td>
<td>n/a</td>
<td>Eq. (10) in [12]</td>
</tr>
<tr>
<td>NMF-DFT</td>
<td>DFTs</td>
<td>Eq. (4.2)</td>
<td>Eq. (4.3)</td>
</tr>
<tr>
<td>NMF-DFT-SC</td>
<td>DFTs</td>
<td>Eq. (4.6)</td>
<td>Eq. (4.7)</td>
</tr>
<tr>
<td>NMF-MEL-SC</td>
<td>DFTs, MELs</td>
<td>Eq. (4.10)</td>
<td>Eq. (4.7)</td>
</tr>
<tr>
<td>NMF-MEL-SC-RC</td>
<td>DFTs, MELs</td>
<td>Eq. (4.10)</td>
<td>Eq. (4.14)</td>
</tr>
</tbody>
</table>
In the first iteration of Eq. (4.5), $H$ was initialized to unity, and the update rule was repeated for 500 iterations. The sparsity penalty factor $\lambda$ was empirically set to 0.1 which is selected according to the performance on the development set.

Both objective and subjective evaluations were conducted to assess the performance of the reference methods discussed above. In both evaluations, the performance of NMF-based methods was first assessed to show the effect of the incremental modifications to the basic exemplar-based sparse representation, namely, NMF-DFT. After that, we compared our proposed NMF-MEL-SC-RC method with the baselines, namely, ML-GMM, DKPLS, DKPLS-DFT and DPLS-DFT methods.

### 4.3.2 Objective evaluations

Mel-cepstral distortion (MCD) calculated between the converted and corresponding reference target Mel-cepstra was employed as an objective evaluation measure. The MCD was calculated as

$$
\text{MCD}[\text{dB}] = \frac{10}{\log 10} \sqrt{2 \sum_{i=1}^{24} (c_i - c_i^{\text{conv}})^2},
$$

where $c_i$ and $c_i^{\text{conv}}$ are the $i^{th}$ coefficients of the reference target and converted MCCs, respectively. For the DKPLS-DFT, PLS-DFT, DPLS-DFT and exemplar-based sparse representation methods, MCCs were computed from the converted spectrograms to calculate the MCD, so that the objective measure was comparable across all the methods. The MCD was calculated frame-by-frame over all the paired frames in the evaluation set, and the average MCD value was reported. A lower MCD value indicates smaller spectral distortion.

#### Effect of dictionary construction

The effect of dictionary construction was examined using NMF-DFT method which represents a basic sparse representation method. In the training set, there were approximately 5500 exemplars from each speaker pair. If all the exemplars were used, the computation and memory usage would be considerably high. Thus, the number of exemplars was varied from 500 to 5500 and observed the conversion performance.
Note that the exemplars were randomly selected from the training set, and a smaller set of exemplars was always a subset of a larger set.

Fig. 4.4 presents the spectral distortions as a function of the number of exemplars, and the results on the development and evaluation sets are presented for comparison. It is observed that the spectral distortion decreases as the number of exemplars increases. With 3000 exemplars, the method achieves almost the same performance as that of 5500 exemplars in terms of spectral distortion. The results on the development set are consistent with those on the evaluation set. Note that 3000 exemplars yield about 2 times faster computation and about half memory usage comparing with 5500 exemplars. Hence \( N = 3000 \) exemplars were used in the following experiments.

![Spectral distortion as a function of the number of exemplars](image)

**Figure 4.4:** Spectral distortion as a function of the number of exemplars \( N \) in a dictionary.

**Analysis of the NMF-based methods**

By setting the number of exemplars in the dictionary to be \( N = 3000 \), the performance of the incremental modifications to the basic exemplar-based sparse representation setup was assessed. The spectral distortions of the NMF-based methods are presented
in Table 4.2, and discussed in details in this Section. Without voice conversion, the spectral distortions between reference source and target MCCs are 7.77 dB and 7.91 dB on the development and evaluation sets, respectively.

Table 4.2: Comparison of spectral distortions of the NMF-based methods

<table>
<thead>
<tr>
<th>Conversion method</th>
<th>Window size</th>
<th>Spectral distortion (dB)</th>
<th>Development</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>No conversion</td>
<td>n/a</td>
<td>7.77</td>
<td>7.91</td>
<td></td>
</tr>
<tr>
<td>NMF-DFT</td>
<td>1</td>
<td>5.48</td>
<td>5.52</td>
<td></td>
</tr>
<tr>
<td>NMF-DFT-SC</td>
<td>1</td>
<td>5.05</td>
<td>5.13</td>
<td></td>
</tr>
<tr>
<td>NMF-MEL-SC</td>
<td>1</td>
<td>5.08</td>
<td>5.18</td>
<td></td>
</tr>
<tr>
<td>NMF-DFT</td>
<td>7</td>
<td>5.25</td>
<td>5.31</td>
<td></td>
</tr>
<tr>
<td>NMF-DFT-SC</td>
<td>7</td>
<td>4.91</td>
<td>4.99</td>
<td></td>
</tr>
<tr>
<td>NMF-MEL-SC</td>
<td>9</td>
<td>4.93</td>
<td>5.03</td>
<td></td>
</tr>
<tr>
<td>NMF-MEL-SC-RC</td>
<td>9</td>
<td>n/a</td>
<td>4.92</td>
<td></td>
</tr>
</tbody>
</table>

Firstly, the effectiveness of spectral compression was examined by comparing the NMF-DFT and NMF-DFT-SC methods. Correspond to Fig. 4.2, spectral distortions as a function of the varied compression factors on both development and evaluation sets are presented in Fig. 4.5. consistent behavior was observed in both development and evaluation sets. A compression factor of 0.4 gives the lowest spectral distortions of 5.05 dB and 5.13 dB on the development and evaluation sets, respectively. Hence $\rho = 0.4$ was chosen as the compression factor to represent NMF-DFT-SC method as shown in Table 4.2 in the reminding experiments. When a compression factor is set to 1.0, NMF-DFT-SC is reduced to the NMF-DFT method.

Secondly, the effect of feature dimensionality reduction in the source dictionary was examined by comparing the NMF-DFT-SC and NMF-MEL-SC methods. As shown in Table 4.2, NMF-DFT-SC method which uses high-resolution source dictionary to estimate the activation matrix produces a spectral distortion of 5.13 dB on the evaluation set. On the other hand, NMF-MEL-SC gives a slightly higher spectral distortion of 5.18 dB on the same set after performing dimensionality reduction.

The computational costs between the NMF-DFT-SC and the NMF-MEL-SC methods was also examined. The computing time was computed over all the testing data, and the average performance for generating one second of speech was reported. The computational costs to generate one second of speech as a function of the window sizes...
Figure 4.5: Spectral distortion as a function of spectral compression factors ($\rho$).

in an exemplar are presented in Fig. 4.6. Note that the computing time was calculated only for the 500 iterations’ multiplicative updates. The MATLAB\textsuperscript{2} codes were run on graphics processing unit (GPU), called GeForce GTX TITAN\textsuperscript{3}. In general, the NMF-MEL-SC method is about 7 times faster than the NMF-DFT-SC method.

Thirdly, the effect of using multiple-frame exemplars was examined to assess whether an exemplar spanning multiple consecutive frames is useful. Fig. 4.7 presents the spectral distortion results as a function of the window size $L$ of an exemplar on both development and evaluation sets. For NMF-MEL-SC method, it is observed that as the window size increases, the spectral distortions consistently decrease and reach their minimum at 9 on both development and evaluation sets. While for NMF-DFT-SC method, spectral distortions have similar trends, but they reach their minimum at 7 and 9 on the development and evaluation sets, respectively. Due the heavy computations, the performance of NMF-DFT-SC when the window sizes were larger than 9 was not tested. When the window size equals to 9, the source dictionary size of

\textsuperscript{2}http://www.mathworks.com/products/matlab/
\textsuperscript{3}http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan
Figure 4.6: Computing time to generate one second of speech as a function of the window size \((L)\) of an exemplar.

NMF-DFT-SC is \(3000 \times (513 \times 9) = 3000 \times 4617\), while that of NMF-MEL-SC is \(3000 \times (50 \times 9) = 3000 \times 450\).

Figure 4.7: Spectral distortion as a function of the window size \((L)\) of an exemplar.
The activation weight for NMF-MEL-SC was analyzed by setting the window size of an exemplar to 9. Fig. 4.8 presents an example of activation weights calculated for a single observation. In the example, there are only 13 exemplars that have weights greater than 0.01, and only two of them have weights greater than 0.05. For further analysis, the average top weights over one utterance was calculated. Given an utterance, the activation matrix was calculated; then the activations corresponding to each source frame were sorted in a descending order; after that, the activations weights were averaged over all the source frames in the utterance. The top 100 averaged weights are presented in Fig. 4.9. It shows that the top 30 or 50 exemplars contribute more to the generated target spectrogram, while the other 2950 exemplars have weights that almost equal to zero. It implies only 1 % or even fewer exemplars are activated in generating each target spectrum, and confirms that effectiveness of the sparsity constraint.

![Exemplar Index vs. Weight](image)

Figure 4.8: Illustration of the activation weights associated to each exemplar to generate a target spectrum.

Lastly, the effectiveness of residual compensation was assessed. The NMF-MEL-SC method using 9-frame exemplars was first applied on the development set to produce the residuals. Then, a mapping between the source and target residuals was established using dynamic partial least squares (DPLS) regression. 20 latent factors were adopted in the DPLS regression based on the performance of DPLS-DFT which was tuned on the development set. After that, at runtime conversion, the same NMF-MEL-SC method was applied to each source utterance to predict a converted spectrogram and at
the same time generate a source residual. The pre-trained DPLS mapping function was applied to the source residual to predict the target residual, which was compensated to the converted spectrogram.

The proposed method was applied on the evaluation set. As shown in Table 4.2, it is observed that residual compensation is able to reduce the spectral distortion from 5.03 dB for NMF-MEL-SC to 4.92 dB for NMF-MEL-SC-RC. It shows the effectiveness of residual compensation.

Overall performance assessment

The proposed NMF-MEL-SC-RC method was compared with the state-of-the-art baselines, namely, ML-GMM method and PLS-based methods. Table 4.3 presents the spectral distortions on the development and evaluation sets. Note that only \( N = 3000 \) exemplars rather than all the exemplars are adopted in the dictionary for the NMF-MEL-SC-RC method, while ML-GMM and PLS-based methods use all the frames \( N \approx 5500 \) in the training set. In the ML-GMM and ML-GMM-GV methods, the number of Gaussian components was set to 32 based on the MCD results on the development set. It is observed that on the evaluation set, the ML-GMM and ML-GMM-GV methods achieve spectral distortions of 5.19 dB and 5.71 dB, respectively, and the
DKPLS method which performs nonlinear mapping gives a spectral distortion of 4.95 dB. Two variants of DKPLS, namely, DKPLS-DFT and DPLS-DFT, are applied on the high-resolution features and produce spectral distortions of 5.13 dB and 5.26 dB, respectively. On the same set, our NMF-MEL-SC-RC method achieves 4.92 dB, which is lower than both the ML-GMM method and the three PLS-based methods.

Table 4.3: Comparison of spectral distortions of the proposed and baseline methods

<table>
<thead>
<tr>
<th>Conversion method</th>
<th>Window size</th>
<th>Spectral distortion (dB)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No conversion</td>
<td>n/a</td>
<td>7.77</td>
<td>7.91</td>
<td></td>
</tr>
<tr>
<td>ML-GMM</td>
<td>3</td>
<td>5.09</td>
<td>5.19</td>
<td></td>
</tr>
<tr>
<td>ML-GMM-GV</td>
<td>3</td>
<td>5.64</td>
<td>5.71</td>
<td></td>
</tr>
<tr>
<td>DKPLS</td>
<td>3</td>
<td>4.88</td>
<td>4.95</td>
<td></td>
</tr>
<tr>
<td>DKPLS-DFT</td>
<td>3</td>
<td>5.07</td>
<td>5.13</td>
<td></td>
</tr>
<tr>
<td>DPLS-DFT</td>
<td>3</td>
<td>5.18</td>
<td>5.26</td>
<td></td>
</tr>
<tr>
<td><strong>NMF-MEL-SC-RC</strong></td>
<td><strong>9</strong></td>
<td><strong>n/a</strong></td>
<td><strong>4.92</strong></td>
<td></td>
</tr>
</tbody>
</table>

The flexibility of the proposed NMF-MEL-SC-RC was then examined by comparing with the DKPLS method. The number of exemplars in the dictionary or the number of frames as training was varied from 500 to 3000. For a fair comparison, the development set for training the residual mapping was also varied accordingly. The spectral distortions on the evaluation set as a function of the number of exemplars/frames is presented in Fig. 4.10. It is observed that the NMF-MEL-SC-RC method has a similar behavior to the NMF-DFT method as shown in Fig. 4.4, and that the effect of the DKPLS method is consistent with [45]. It is worth noting that NMF-MEL-SC-RC is more stable than the DKPLS method even when the training data is limited. Note that the exemplars/frames were randomly selected from the training set and high-resolution features for NMF-MEL-SC-RC were paired with MCCs used in DKPLS.

In general, the proposed NMF-MEL-SC-RC method robustly achieves lower spectral distortions with varying the training data, and also works well on high-resolution features. PLS-based methods give lower spectral distortions on low-resolution features, but the performance drops considerably when applied to high-resolution features.
Chapter 4. Exemplar-based sparse representation with residual compensation for voice conversion

Figure 4.10: Spectral distortions on the evaluation set as a function of the number of exemplars/frames $N$ as training.

4.3.3 Subjective evaluations

Listening tests were conducted to compare the performance between the proposed NMF-MEL-SC-RC and the baseline methods in terms of speech quality and speaker individuality. Amazon Mechanical Turk (AMT)\footnote{https://www.mturk.com}, a crowd sourcing platform, was used in each listening test. The same platform has been earlier used in subjective evaluations, e.g. in [115–117]. In the evaluation set, there were 12 conversion pairs and each pair had 20 utterances. Hence, in total there were 240 ($12 \times 20$) converted utterances. In the listening test, 20 utterances were randomly selected from the 240 utterances for each listener (or worker as called in AMT) to avoid bias on utterances. Moreover, 3 golden standard pairs were randomly mixed with the 20 real testing utterances to prevent cheating as advised in [117]. In each test, ten paid listeners were involved.

Analysis of the NMF-based methods

Four preference listening tests were conducted to assess the effect of the incremental modifications to the basic sparse representation setup. This section focused on speech
quality in this section.

Firstly, an AB preference evaluation was performed between NMF-DFT and NMF-DFT-SC methods regarding the effect of spectral compression. In an AB preference test, speech samples converted by two methods, namely NMF-DFT and NMF-DFT-SC, were presented to listeners in a random order, and the listeners were asked to choose the one that sounded more natural. Fig. 4.11 presents the preference scores. It shows that the preference scores are consistent with the spectral distortions and NMF-DFT-SC achieves significantly better speech quality than NMF-DFT without spectral compression. In general, both objective and subjective evaluations confirm the effectiveness of spectral compression.

![Preference test results of speech quality with 95% confidence intervals for NMF-DFT and NMF-DFT-SC methods.](image)

Secondly, a three-way preference test was conducted to examine the effect of the feature dimensionality reduction in the source dictionary. Different from the AB preference test, the three-way test had three options. Two speech samples generated from NMF-DFT-SC and NMF-MEL-SC methods were randomly presented to each listener, and then the listeners were asked to decide which sample is more natural. If they were not able to perceive the difference between two samples, they were asked to choose the option claiming no preference. The preference test results are presented in Fig. 4.12. It is observed that around 60% sample pairs have the same speech quality, while NMF-DFT-SC achieving around 25% preferences is slightly better than NMF-MEL-SC of around 15%, but the difference is not statistically significant. Again, note that NMF-MEL-SC was around 7 times faster than NMF-DFT-SC and that there was 10 times memory reduction of NMF-MEL-SC comparing with NMF-DFT-SC. We hence conclude that NMF-MEL-SC with slightly performance drop is computationally more
Chapter 4. Exemplar-based sparse representation with residual compensation for voice conversion

efficient than NMF-DFT-SC.

Figure 4.12: Preference test results of speech quality with 95 % confidence intervals for NMF-DFT-SC and NMF-MEL-SC methods.

Thirdly, an AB preference test was conducted to assess the effectiveness of using multiple-frame exemplars in the source dictionary. NMF-MEL-SC using a single frame as an exemplar was compared with that using a nine-frame speech segment as an exemplar. Fig. 4.13 presents the subjective evaluation results. It shows that NMF-MEL-SC using nine-frame exemplars is significantly better than that using single-frame exemplars, and confirms the effectiveness of using multiple-frame exemplars. The subjective results are consistent with the objective spectral distortions.

Figure 4.13: Preference test results of speech quality with 95 % confidence intervals for NMF-MEL-SC with and without multiple-frame exemplars.

Lastly, an AB preference test was conducted to examine the effectiveness of residual compensation by comparing NMF-MEL-SC and NMF-MEL-SC-RC. Fig. 4.14 presents the preference scores. It is observed that NMF-MEL-SC-RC achieves a significantly higher preference score than that of the NMF-MEL-SC method. The preference results are consistent with the objective evaluations. In general, NMF-MEL-SC-RC achieves a lower spectral distortion and better than the NMF-MEL-SC method.

73
Overall speech quality assessment

Subjective evaluations were conducted to assess the speech quality between the proposed and baseline methods.

Firstly, AB preference tests were conducted to assess the speech quality between the proposed NMF-MEL-SC-RC method and PLS-based methods, where PLS-based methods used the whole training set $N \approx 5500$, while the proposed method used $N = 3000$ exemplars. Fig. 4.15 presents the preference test results. In Fig. 4.15a, it is observed that the proposed NMF-MEL-SC-RC achieves similar performance to DKPLS method, in the sense that each method’s preference score falls into the other method’s confidence intervals. Figs. 4.15b and 4.15c show that the proposed method is significantly better than both DKPLS-DFT and DPLS-DFT methods. As the three PLS-based methods were compared with the same NMF-MEL-SC-RC method, the preference scores imply that even though DKPLS works well with low-resolution MCC features, the performances of DKPLS as well as DPLS are degraded considerably in face of high-dimensional features. We conclude that PLS-based methods are not as flexible as our NMF-MEL-SC-RC method in handling high-dimensional features. The preference scores are consistent with the spectral distortions as shown in Table 4.3.

Secondly, an AB preference test was conducted to assess the flexibility of the NMF-MEL-SC-RC and DKPLS methods. Here, both methods used 500 exemplars/frames as training. Fig. 4.16 presents the preference scores with 95% confidence intervals. It is observed that the proposed NMF-MEL-SC-RC method is slightly better than the DKPLS method when limited training data are available, but the difference is not statistically significant. Even though previous result with 5500/3000 exemplars...
Chapter 4. Exemplar-based sparse representation with residual compensation for voice conversion

Figure 4.15: Preference scores of speech quality with 95% confidence intervals for PLS-based methods with $N \approx 5500$ training frames and our proposed NMF-MEL-SC-RC method with $N = 3000$ exemplars in the dictionary. From top to bottom, (a), DKPLS vs. NMF-MEL-SC-RC; (b), DKPLS-DFT vs. NMF-MEL-SC-RC; and (c), DPLS-DFT vs. NMF-MEL-SC-RC.

shows that the two methods achieve almost the same performance, the result using 500 exemplars shows the superiority of the proposed NMF-MEL-SC-RC. It confirms the flexibility of the proposed NMF-MEL-SC-RC, and it is consistent with the spectral distortions as presented in Fig. 4.10.

Next, a mean opinion score (MOS) test was conducted to compare the NMF-MEL-SC-RC and ML-GMM methods, as the ML-GMM method is a well-established baseline. The opinion score was set to a five-point scale: 1 = bad, 2 = poor, 3 = fair, 4 = good and 5 = excellent. 20 speech pairs were randomly selected from the 240 pairs, and each pair consisted of two conversion samples from the ML-GMM and NMF-MEL-SC-RC methods. The language content of the paired conversion samples was exactly the
same. 3 pairs with natural speech were mixed with the 20 real testing pairs as golden standard pairs to exclude cheaters. The average MOS results with 95% confidence intervals are presented in Fig. 4.17. ML-GMM has a MOS of 2.49, while NMF-MEL-SC-RC achieves 3.15. It clearly shows that the proposed NMF-MEL-SC-RC method significantly outperforms the baseline ML-GMM method.

Finally, another MOS test was conducted to compare the NMF-MEL-SC-RC and ML-GMM-GV methods. It is generally believed that the ML-GMM-GV method is able to avoid the over-smoothing problem in conventional statistical parametric methods, and produces natural converted speech. We followed the same procedures as that for comparing the NMF-MEL-SC-RC and ML-GMM methods. Fig. 4.18 presents the average MOS results with 95% confidence intervals. It is observed that ML-GMM-GV achieves a MOS of 3.07, while NMF-MEL-SC-RC gives 2.95. Although the average MOS of ML-GMM-GV is slightly higher than NME-MEL-SC-RC, the ML-GMM-GV
method does not outperform our proposed NMF-MEL-SC-RC method significantly, in the sense that the MOS of each method falls into the other method’s confidence intervals. We note that the scores in Figs. 4.17 and 4.18 cannot be compared directly, as the two tests were conducted independently and the listeners may not be exactly the same for both tests.

Figure 4.18: Mean opinion scores with 95% confidence intervals for ML-GMM-GV and NMF-MEL-SC-RC methods.

Identity evaluations

Subjective evaluations were conducted to assess the speaker similarity/individuality performance. From our previous experience [20], speech quality would affect the speaker individuality evaluation results in a preference test, as the listeners usually paid more attention on speech quality and preferred to choose samples that sound more natural. To this end, we conducted an XAB test for each method independently for fair comparison. In the test, 20 pairs were presented to the listeners, and each pair consisted of three speech samples: X = a converted sample, A = a source sample and B = a target sample. The language content of paired A and B was the same while that of X was different to make sure the listeners focus on the spectral attributes other than prosodic patterns. During the test, the converted sample was first presented as a reference. Then, source and target samples were presented in a random order. The listeners were asked to decide whether sample A or B sounded closer to X in terms of speaker individuality. The identification rate, which is the percentage of converted samples identified as target, was reported. Similar to [34, 45], all the inter-gender conversion pairs were identified correctly in an initial listening test. We hence only reported the results of intra-gender conversion, which was a more challenging task than the inter-gender task.
Fig. 4.19 presents the identification rate results. The results suggest that the NMF-based methods achieve slightly higher identification rates than the baseline ML-GMM method, and similar identification rates to the ML-GMM-GV method. In particular, NMF-MEL-SC-RC achieves an identification rate of 79.50 % while ML-GMM and ML-GMM-GV reaches 73.50 % and 80.00 %, and DKPLS attains 77.00 %. It is also observed that DPLS-DFT gives the lowest identification rate of 67.00 %. We note that the ML-GMM-GV achieves the smallest variance of the identification rates from all the listeners.

In general, both the speech quality and identity tests confirm the effectiveness and flexibility of our proposed NMF-MEL-SC-RC method over a large set of state-of-the-art baseline methods. The subjective results are consistent with the objective spectral distortions.
4.4 Discussion

The experimental results have confirmed the effectiveness of the proposed exemplar-based sparse representation with residual compensation framework. Even though it is a nonparametric framework, there are some fundamental relationships between the proposed NMF-MEL-SC-RC method and the state-of-the-art methods as discussed in this section.

4.4.1 Relationship with vector quantization and frame selection methods

The vector quantization (VQ) and frame selection are two related methods that use original feature vectors to generate the target speech. In the VQ method, a codebook is established as a subset of the source-target feature pairs. At runtime conversion, the codebook is employed to find a target feature whose paired source feature is close to the given source feature. A similar method named frame selection [93], also called unit selection [51], establishes a source-target correspondence during offline training. At runtime conversion, the correspondence is applied to select a target feature vector given a source feature vector.

The advantage of using the VQ and frame selection is that the selected target feature keeps the spectral details as original one. However, the hard clustering nature of VQ results in the incoherence phenomenon across frames. This phenomenon is also usually observed in the frame selection method [118]. On the other hand, in the proposed exemplar-based sparse representation method, exemplars in the paired source-target dictionaries can be treated as entries of a codebook in the VQ method, or source-target correspondence in the frame selection method. The proposed exemplar-based sparse representation with residual compensation method differs from the VQ and frame selection methods by presenting an observation as a linear combination of exemplars, whereas the VQ and frame selection methods calculate the selection costs for exemplars/frames independently from each other.

In addition, the frame selection method usually requires a significant amount of training data, while the proposed method works with even 500 exemplars, that is 2.5 seconds of speech. Similar to the work in [93], the proposed method can also be
combined with the frame selection method, in the way that the converted speech by the proposed method is used as the reference target speech for frame selection.

### 4.4.2 Relationship with JD-GMM methods

In JD-GMM based methods, the joint mean vectors $\mu_k^{(z)}$ and covariance matrices $\Sigma_k^{(z)}$ are employed to establish a conversion function to predict a target feature $\hat{y}$ given a source feature $x$:

$$
\hat{y} = \sum_{k=1}^{K} p_k(x) (\mu_k^{(y)} + \Sigma_k^{(yx)} (\Sigma_k^{(xx)})^{-1} (x - \mu_k^{(x)})),
$$

(4.16)

$$
\mu_k^{(z)} = [\mu_k^{(x)}; \mu_k^{(y)}] = \sum_{n=1}^{N} z_n \cdot \gamma_{n,k},
$$

(4.17)

$$
\Sigma_k^{(z)} = \begin{bmatrix}
\Sigma_k^{(xx)} & \Sigma_k^{(xy)} \\
\Sigma_k^{(yx)} & \Sigma_k^{(yy)}
\end{bmatrix} = \sum_{n=1}^{N} \gamma_{n,k} \cdot (z_n - \mu_k^{(z)}) (z_n - \mu_k^{(z)})^\top,
$$

(4.18)

where $\gamma_{n,k}$ is the occupation probability of the $n^{th}$ frame belonging to the $k^{th}$ Gaussian component [34, 59], and $p_k(x)$ is the posterior probability of the source feature $x$ generated from the $k^{th}$ Gaussian component.

On the other hand, as presented in Eq. (4.14), if the residual compensation is performed on a linear amplitude scale and the compression factor $\rho$ is set to 1.0, the predicted target feature $\hat{y}_{(DFT)}$ can be presented as

$$
\hat{y}_{(DFT)} = B_{(DFT)} h + F(r^{(X)}),
$$

(4.19)

where $r^{(X)}$ is one column of $R^{(X)}$.

Comparing Eqs. (4.16), (4.19) and (4.17), it is observed that both $B_{(DFT)} h = \sum_{n=1}^{N} b_n^{(DFT)} \cdot h_n$ and $\mu_k^{(y)} = \sum_{n=1}^{N} y_n \cdot \gamma_{n,k}$ do conversion as a linear combination of either $b_n^{(DFT)}$ or $y_n$. Note that $b_n^{(DFT)}$ is a high-resolution spectrum, and $y_n$ is the corresponding low-resolution MCC feature. The activation vector $h$ is constrained to be sparse, while $\gamma_k = [\gamma_{1,k}, \gamma_{2,k}, \ldots, \gamma_{n,k}, \ldots, \gamma_{N,k}]$ does not have such a constraint. Thus, $\gamma_k$ interpolates training samples to generate the mean vectors, in some sense, it tries to use as many as possible training samples to represent an unseen sample;
while the sparse representation method uses a minimum number of samples to describe the unseen sample. For example, if a testing sample is included in the training, sparse representation similar to the VQ method is able to find the exact sample, while GMM-based approaches interpolate the whole training sample space. In this way, our proposed exemplar-based sparse representation with residual compensation method is able to avoid the over-smoothing effect introduced by the statistical average.

Moreover, as discussed in [119], the entries in $h$ are conditionally dependent on each other given the dictionary and observation. However, the entries in $\gamma_k$ are dependent only through the scalar normalization constant, otherwise they are independent from each other. Thus, $h$ is able to benefit from the dependencies of exemplars for regression, while $\gamma_k$ cannot.

### 4.4.3 Relationship with PLS

In partial least squares regression, given parallel data $X \in \mathbb{R}^{d \times N}$ and $Y \in \mathbb{R}^{d \times N}$, we have such decompositions:

$$X \approx OP, \quad (4.20)$$
$$Y \approx VQ, \quad (4.21)$$

where $P \in \mathbb{R}^{p \times N}$ and $Q \in \mathbb{R}^{p \times N}$ are projections or factor matrices of $X$ and $Y$, respectively, and $O \in \mathbb{R}^{d \times p}$ and $V \in \mathbb{R}^{d \times p}$ are loading matrices corresponding to $P$ and $Q$, respectively.

As discussed in [120], a transformation matrix $T$ can be estimated from Eq. (4.20) and (4.21), and applied to a given source feature $x$ for predicting a target feature $\hat{y}$ during runtime as

$$\hat{y} = Tx + T(\mu^{(Y)} - \mu^{(X)})$$
$$= YP^\top (QXX^\top P^\top)^{-1}QXx + T(\mu^{(Y)} - \mu^{(X)}), \quad (4.22)$$

where $\mu^{(X)}$ and $\mu^{(Y)}$ are the mean vectors of source and target training data, respectively.

Eq. (4.22) has a considerable similarity to Eq. (4.19), if we treat $P^\top (QXX^\top P^\top)^{-1}QXx$ as the activation vector $h$, and $Y$ as the target dictionary. The proposed method is
hence similar and comparable to DPLS-DFT, which also operates on high-resolution features. The difference between the proposed method and DKPLS/DKPLS-DFT is that DKPLS methods perform the spectral mapping in kernel space which introduces nonlinearity, while the proposed method does not.

There are two advantages of the proposed method over PLS-based methods. First, the flexibility allows the proposed method to be robust in handling high-resolution features and in face of limited training samples. The experimental results show that in face of high-resolution features and limited training samples, the performance of PLS-based methods is degraded considerably, while the proposed method is still reliable in such scenario. Second, the experimental results also confirmed the scalability of the proposed method, in the sense that the performance of the proposed method can be boosted by simply appending new exemplars to the dictionary without much tuning. However, for PLS-based methods, re-optimization is required in the face of new training data.

### 4.4.4 Computational complexity and memory footprint

In comparison to the reference methods, major drawbacks of the proposed method are the computational complexity and memory footprint. From the perspective of computational complexity, to generate one second of target speech, the ML-GMM and DKPLS methods take 0.41 and 0.06 seconds, respectively, on a 2.3 GHz Intel i7 core when implemented in MATLAB. However, the proposed method costs 19.02 seconds. The computational cost of the proposed method is about 45 times higher than the ML-GMM method and about 295 times higher than the DKPLS method.

For the memory footprint, at runtime, the ML-GMM method only needs to store $32 + 2 \times 32 \times (48 + (48 + 24)) = 7712$ parameters, supposing 32 Gaussian mixtures, 24 dimensional MCCs, and static+delta coefficients. The DKPLS method only requires to store $(200 \times 3) \times 24 = 14400$ parameters. However, the proposed method needs to store the source and target dictionaries. The size of the target dictionary is $513 \times 3000$, and that of the source dictionary is $9 \times 50 \times 3000$. In total, the dictionaries’ size is $513 \times 3000 + 9 \times 50 \times 3000 = 2889000$. Hence, the memory occupation of the proposed method is about 375 times higher than the ML-GMM method, and is about 200 times higher than the DKPLS method.
4.5 Conclusions

This chapter proposed an exemplar-based sparse representation framework as an alternative nonparametric framework for voice conversion. The flexibility of this framework allows us to easily adapt it to new training data, and makes it more robust in handling high-resolution spectra directly to maintain spectral details for better speech quality. In addition, the use of coupled dictionaries avoids to estimate the correlation/covariance matrix which is required in conventional statistical methods and is problematically estimated when source-target feature pairs have a low correlation [15]. The experimental results confirmed the effectiveness of the proposed exemplar-based sparse representation with residual compensation method, which achieves a spectral distortion of 4.92 dB, a MOS of 3.15 and a speaker identification rate of 79.50 %, outperforming the baseline ML-GMM method which gives a spectral distortion of 5.19 dB, a MOS of 2.49 and a speaker identification rate of 73.50 %. Moreover, the proposed method is also more flexible than PLS-based methods, and comparable with the ML-GMM-GV method.

The main findings of this chapter are:

- Sparse representation is able to produce relatively high quality speech. It allows us to model high-resolution features directly for spectral details, and the activation vector for regression is constrained to be extremely sparse, in this way, the over-smoothing effect can be avoided.

- Spectral compression is helpful. A compression factor is able to control the spectrum intensity and affect the estimation of activations as well as the spectrogram generation.

- Multiple-frame exemplars which are able to describe the time sequence structure of speech are beneficial to reduce spectral distortion and to produce better speech quality.

- Residual compensation works well to reduce spectral distortion and to enhance speech quality. The sparse representation modeling capacity can be boosted by compensating source model residuals to the converted spectrograms.
As an alternative framework, the proposed exemplar-based sparse representation method is complementary with the statistical parametric methods, which can be employed to perform residual compensation.

Currently, parallel data, which is not always available, is required to construct source-target dictionaries. It would be interesting to find a method to relax such a constraint. Moreover, the computation of exemplar-based sparse representation is considerably higher than the ML-GMM method. It is possible to reduce computational time by applying low-resolution features to estimate the activation matrix and by using a small set of exemplars in the dictionaries. Those directions will be continued in near future as a follow-up work.
Chapter 5

Vulnerability Evaluation of Speaker Verification Under Voice Conversion Spoofing Attacks

In Chapter 3 and 4, novel voice conversion approaches have been proposed to improve the speaker similarity and speech quality for human listening. As discussed in Chapter 2, the natural application of voice conversion is to personalise text-to-speech synthesis, which targets for human ears. By changing the speaker identity, voice conversion technology can also be employed to spoof automatic speaker verification systems, the objective of which is to automatically accept or reject a claimed identity given a speech sample.

This chapter uses voice conversion technology as a spoofing attack approach to systematically examine the vulnerability of speaker verification systems. The study is important, as when deploying a speaker verification system, a major concern is whether it is still robust against spoofing attacks. By understanding the weak links in speaker verification systems, countermeasures can be developed to enhance the security of speaker verification systems. This work has been published in the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2012 [22], Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC) 2012 and 2013 [6,23], Interspeech 2013 [24].

The rest of this chapter is organised as follows: Section 5.1 presents the motivation
of this study; Section 5.2 discusses the related work on the evaluation of vulnerability of speaker verification systems; Section 5.3 presents the voice conversion system used in this study; Section 5.4 briefly introduces the speaker verification systems used in this study; Section 5.5 presents experimental protocol, including the evaluation measures and the dataset design; Section 5.6 discusses the experimental setups, results and analysis; Section 5.7 concludes this chapter and presents the findings from this study.

5.1 Motivation

Speaker verification [4, 121] is a low-cost biometric solution. Unlike other forms of biometrics, such as fingerprint or iris recognition, a speech sample can be acquired remotely using existing landline, cellular and voice-over IP communication channels without additional hardware. Currently, speaker verification has already been deployed in applications like real-time caller verification [122] and smartphone login [123] to validate transactions in e-commerce and to safeguard personal information.

When deploying a speaker verification system, the system is expected to be accurate to regular clients, and also robust against spoofing attacks. A typical speaker verification system is optimised to accept genuine speakers and to reject impostors, assuming natural human speech. However, in practice, techniques such as voice conversion can be adopted as a spoofing attack approach to fool speaker verification systems. As pointed out in [124], spoofing attacks can take place at two locations in a speaker verification system: at the microphone sensor and during the transmission of the acquired speech signal. At the sensor level, an impostor, also called an adversary, could compromise the system by replaying a pre-recorded speech signal or impersonating the target speaker at the sensor. During the transmission, the acquired speech signal could be replaced by a falsifying one. Generally, a spoofing attack is to employ a forged signal as the system input in either of the above two locations.

A speaker verification system makes a decision based on the feature distributions through speaker modelling. The feature are generally extracted at three different levels to describe speaker individuality [4, 125]: a) short-term spectrum; b) prosody; and c) high-level idiolectical/lexical features. Being information-rich and easy to compute, spectral features are the primary features used by modern classifiers. Adding prosodic features may further enhance accuracy [126–128].
On the other hand, voice conversion systems also operate on short-term spectrum and prosody features in order to mimic a target speaker’s voice. It is hence able to move an impostor’s spectral feature and prosodic feature distributions towards those of a genuine target speaker’s distribution and therefore presents a great risk to speaker verification systems.

Earlier studies on the vulnerabilities of text-independent and text-dependent speaker verification systems usually include only one classifier and spoofing method. With varying protocols and evaluation datasets across these studies, it is difficult to draw some general conclusions or recommendations. While many believe that text-dependent systems, being linguistically constrained, should be more robust against intentional spoofing, to date little evidence exists to support this claim.

To address the above questions, this work focuses on the vulnerabilities of modern speaker verification systems in the face of spoofing attacks simulated by voice conversion. In particular, two standard databases, the National Institute of Standards and Technology (NIST) speaker recognition evaluation (SRE) 2006\(^1\) and RSR2015\(^2\), are employed. The performance of both text-independent and text-dependent speaker verification systems are also examined under the same spoofing attack, which is simulated by the mainstream joint density Gaussian mixture model voice conversion. Seven text-independent speaker verification systems are evaluated on the NIST SRE 2006 database. This includes three lightweight classifiers, GMM-UBM [129], VQ-UBM [130] and GLDS-SVM [131], three more advanced classifiers, GMM-SVM [131], joint factor analysis (JFA) [132] and the state-of-the-art i-vector PLDA [133], and one fusion system taking advantage of the previous 6 classifiers. Two variants of text-dependent systems are evaluated as well on the RSR2015 databases. In addition, the voice conversion performance is also employed to explain the vulnerabilities of speaker verification under voice conversion spoofing.

\(^1\)http://www.itl.nist.gov/iad/894.01/tests/sre/2006/index.html
\(^2\)http://www.signalprocessingsociety.org/technical-committees/list/sl-tc/spl-nl/2012-05/the-rss2
5.2 Related work on speaker verification vulnerabilities

The most common spoofing technique is replay attack – the rendering of previously recorded target speaker utterances [134,135]. Such attack is effective in spoofing text-independent systems but not viable for generating utterances of specific content or maintaining a live conversation in call-center applications. Aside from replay attacks, impersonation by human beings has also received attention in [136–138]. This attack has occasionally been successful in spoofing speaker verification systems. However, impersonators tend to mimic prosody, pronunciation and lexicon rather than the spectral cues used by speaker verification systems. Apparently, there are other better ways of attack as facilitated by recent advances in speech processing to spoof speaker verification systems.

Speech synthesis and voice conversion are two highly flexible ways to generate speech with a decent target speaker voice quality. They represent an emerging threat to the security of speaker verification systems. Having enough technical skills, one can easily produce speaker adapted voices using tools such as Festival\(^3\) and other commercial and open-source solutions that will be available in the not-so-distant future. Indeed, unit selection [92], statistical parametric [139] and hybrid [140] synthesis methods are able to generate speech adapted to a target speaker with an acceptable quality. Generally speaking, a modern statistical parametric synthesis technique first trains an average voice model from large corpus, which is subsequently adapted to a specific target speaker using a small amount of adaptation utterances [141].

Voice conversion [68] attempts to achieve the same effect as human impersonation and adapted speech synthesis, but operates on speech signal itself. Most voice conversion techniques do not require transcriptions, prosody prediction, or additional off-line corpora. Both speech synthesis [142–145] and voice conversion techniques [22,23,146–151] have been shown to increase the error rates of state-of-the-art classifiers to an unacceptable level. Voice conversion [68] has attracted increasing interest in the context of ASV spoofing for over a decade [152]. In [152], the vulnerability of a GMM-UBM ASV system was evaluated using the YOHO corpus, which consists of 138 speakers.

\(^3\)http://festvox.org/index.html
Experimental results showed that the FAR increased from 1.45 % to 86 % as a result of voice conversion attacks.

Some of the work relevant to text-independent speaker verification spoofing includes that in [147,153]. The work in [153] evaluated the vulnerability of a GMM-UBM ASV system. Experiments reported on the 2004 NIST speaker recognition evaluation (SRE) dataset showed that a baseline EER of 16 % increased to 26 % as a result of voice conversion attacks. The work in [147] investigated a Gaussian-dependent filtering approach to convert the spectral envelope of the input speech signal towards that of the target speaker. These experiments, conducted on the 2005 NIST SRE dataset showed that the baseline EER for a GMM-UBM system increased from 8 % to over 60 % as a result of voice conversion attacks which exploit knowledge of the ASV system. The work in [148], conducted on the 2005 and 2006 NIST SRE datasets, showed a much reduced degradation of EER from 6.61 % to 28.07 % when different feature parameterisations are used for ASV and voice conversion.

The work in [22] and [23] extended the study of GMM-UBM systems to consider an array of different approaches to ASV. The work was performed on the 2006 NIST SRE dataset using both joint-density GMM and unit selection approaches to voice conversion. Even if converted speech could be detected easily by human listeners, experiments involving six different ASV systems showed universal susceptibility to spoofing. The FAR of the JFA system increased from 3.24 % to over 17 % when a GMM-based voice conversion attack happened, and the FAR of even the most robust PLDA system increased from 2.99 % to over 40 % in face of an unit selection conversion attack. Such result is due to the considerable overlap in the distribution of ASV scores for genuine and converted speech.

Still in the context of text-independent ASV, other work relevant to voice conversion includes attacks referred to as artificial signals. It was noted in [154] that certain short intervals of converted speech yielded extremely high scores or likelihoods. Such intervals are not representative of intelligible speech but are nonetheless effective in overcoming ASV systems which lack any form of speech quality assessment. Artificial signals optimised with a genetic algorithm were shown to increase in EER from 10 % to almost 80 % for a GMM-UBM system and from 5 % to almost 65 % for a factor analysis (FA) based system.

The work in [155] examined the vulnerability of several state-of-the-art text-dependent
systems, namely, i-vector, GMM-NAP and HMM-NAP systems. Among the three systems, HMM-NAP employed a speaker-independent hidden Markov model (HMM) instead of a GMM to capture temporal information. Results showed that voice conversion increases the EERs and FARs of all the three systems. Specifically, the EER and FAR of the most robust HMM-NAP system increased from both 1.00% to 2.90% and 36.00%, respectively.

This chapter focuses on voice conversion spoofing, which presents a practical threat to both text-independent and text-dependent speaker verification systems. The framework used in this study is presented in Fig. 5.1. This study follows the standard speaker verification architecture which is supposed to take only natural voice as input, and add a voice conversion system at the input point which converts one’s voice to another to create spoofing attacks. In a genuine trial, a genuine voice goes directly to the feature extraction module, while in an impostor trial, an impostor’s voice passes through the voice conversion to impersonate the target genuine speaker.

Figure 5.1: Illustration of the vulnerabilities evaluation framework used in this study.

### 5.3 Voice conversion system

In the spoofing attack study, it is expected that the voice conversion technique is able to convert the speaker identity efficiently. It is also expected that the voice conversion technique is computational efficient, as a considerable number of conversion pairs are required in the vulnerability evaluation. To this end, this chapter decides to employ
the classic joint-density Gaussian mixture model (JD-GMM) method to conduct spoofing attacks. In Chapter 4, several voice conversion methods have been compared on the same dataset through objective spectral distortion and subjective listening tests. It is observed that the joint-density Gaussian mixture model (JD-GMM) method is able to achieve comparable performance in terms of speaker similarity comparing with other conversion methods and the exemplar-based methods proposed in Chapter 4. More important, the JD-GMM method is computational efficient comparing with other methods.

The JD-GMM conversion approach is illustrated in Fig. 5.2. The JD-GMM system consists of two processes, an offline training process and a runtime conversion process. During the training process, feature representations are extracted from source and target speech signals in the form of parametric vectors and then a conversion function between source and target speech is established through Gaussian mixture model. At runtime, the conversion function transforms the source features extracted from a source speech signal to generate a converted speech sample. More details on JD-GMM approach can be found in Chapter 2.

Figure 5.2: Diagram of a joint density Gaussian mixture model (JD-GMM) based voice conversion system.

In the spoofing attack scenario, the source speaker is the attacker, while the target speaker is the genuine speaker that the attacker attempts to mimic.
5.4 Speaker Verification Systems

Speaker verification systems typically operate in one of two input modes: *text-independent* or *text-dependent*. Text-independent methods assume free text, while text-dependent systems enforce or prompt the user to speak a known pass-phrase. Text-independent systems are often used for off-line screening, indexing or forensic uses, involving non-co-operative users. Text-dependent systems requiring co-operative users [156, 157], in turn, are commonly used for authentication applications.

Since speaker verification usually takes place under remote scenarios without a face-to-face contact with a human operator, spoofing becomes a fundamental concern. It is understood that there is no absolutely safe biometrics. A biometric authentication method can always be intentionally circumvented or spoofed [158]. As highlighted in the recent review [159], the development of protocols and countermeasures for speaker verification lags behind that for other biometric systems. The goal of this work is to provide a better understanding of the vulnerabilities of both text-independent and text-dependent systems.

In this study, both text-independent and text-dependent speaker verification systems are considered. The text-independent systems only rely on spectral cues while text-dependent systems take into account the language content as additional information. Two variants of text-dependent systems are explored to analyse the effect of temporal information.

5.4.1 Text-independent speaker verification systems

In a text-independent speaker verification system, there is no text constraint during both off-line enrolment and run-time verification. Seven text-independent speaker verification systems are implemented for comparison. These systems are briefly introduced as follows.

- **GMM-UBM**: This is the standard Gaussian mixture model (GMM) with universal background model (UBM) [129]. The target speaker models are obtained by adapting the UBM with maximum *a posteriori* (MAP) adaptation [129, 160].

- **VQ-UBM**: Similar to GMM-UBM, we model each speaker using an adapted vector quantisation codebook [130].
• **GLDS-SVM:** In this system, generalised linear discriminant sequence (GLDS) kernel first maps each feature sequence to a polynomial supervector explicitly, and then a support vector machine (SVM) [131] with a linear kernel is used for classification. Speaker models are trained using LibSVM [161].

• **GMM-SVM:** SVM is employed for classifying GMM supervectors [131], and the session variability is compensated by nuisance attribute projection (NAP) [162]. Same as GLDS, LibSVM [161] with linear kernel scoring is adopted.

• **GMM-JFA:** Joint factor analysis (JFA) model [132] decomposes a GMM supervector into speaker-independent, speaker-, and channel-dependent components for intersession and speaker variability compensation.

• **PLDA:** Probabilistic linear discriminant analysis (PLDA) [100, 133] is a generative model which is similar to the GMM-JFA model. PLDA model operates on i-vectors other than GMM supervectors.

• **Fusion system:** Fusion system is a commonly used in speaker verification to take advantage of multiple sub-classifiers [163, 164]. To this end, linear fusion weights for the previous 6 systems are trained on the baseline corpus using logistic regression model [165] as implemented in the FoCal toolkit⁴.

### 5.4.2 Text-dependent speaker verification systems

The effect of spoofing attacks against text-dependent systems are also investigated. To allow a tractable analysis, this study uses a hierarchical acoustic modelling [166, 167] as shown in Fig. 5.3, in which two variants of text-dependent speaker models are progressively trained from the same universal background model, according to the formulation of the maximum a posterior (MAP) adaptation [129, 160].

Two text-dependent classifiers are considered as follows:

• **TD-GMM:** In this setup, a speaker- and text-dependent GMM model is adapted from a universal background model (UBM). The top and middle layers in Fig. 5.3 hence correspond to the good old GMM-UBM [129]. In practice, GMMs are

⁴https://sites.google.com/site/nikobrummer/focal
pass-phrase dependent, so this work refers to this GMM-UBM model as GMM-based text-dependent model (TD-GMM).

- **TD-HMM**: In the bottom layer, a speaker-dependent and sentence-level hidden Markov model (HMM) is adapted from the middle layer TD-GMM. In particular, each state of the HMM is a GMM adapted from the TD-GMM of the speaker by using the MAP criteria. We hence call this pass-phrase and speaker-dependent HMM approach as HMM-based text-dependent model (TD-HMM).

In this study, the structure in Fig. 5.3 is considered to present TD-GMM and TD-HMM for two reasons. Firstly, the RSR2015 text-dependent speaker verification database consists of utterances with very short duration (3 seconds of nominal speech, see Section 5.5.2). For short-duration training and test utterances, the conventional GMM-UBM with MAP adaptation has shown to perform equally well as compared to JFA or PLDA [168]. Second, a more tractable analysis is possible given that the bottom layer HMM models additional temporal information absent in the middle layer GMM.

The likelihood ratio of TD-GMM is calculated between $\lambda_{UBM}$ and $\lambda_{TD-GMM}$, and, similarly, the likelihood score of TD-HMM is obtained between $\lambda_{UBM}$ and $\lambda_{TD-HMM}$.
Chapter 5. Vulnerability Evaluation of Speaker Verification Under Voice Conversion Spoofing Attacks

5.4.3 Points of speaker verification vulnerabilities

A speaker verification system makes a decision based on the feature distributions through speaker modelling [4]. The feature extraction and speaker modelling modules are hence the two most important components. Accordingly, there are two classes of weak links, one in feature extraction and the other in speaker modelling.

From the perspective of feature representation, it is known to the public that speaker verification systems use spectral, prosodic and linguistic features. Thus, speaker verification systems may be vulnerable to the attackers that can manage to mimic those features. On the other hand, voice conversion can modify or mimic all the three levels of features that are also used in speaker verification. Given a sequence of features from an attacker, voice conversion technology can project the attacker’s features to the target speaker’s feature space through the mapping function, and in this way, the speaker verification systems can be deceived by the generated target features.

The spectral and prosodic features are popular features used in speaker verification. In particular, due to the simplify and robust performance, the spectral features are widely used. MFCCs, LPCCs, and LSFs are the popular features to describe the spectral attributes, while F0, intensity, duration and intonation are shared by a large range of speaker verification systems to represent prosodic attributes. On the other hand, those spectral and prosodic features are also involved in voice conversion. Therefore, knowing how spectral or prosodic features are used in a speaker verification system, one is able to devise a spectral prosodic mapping that generates spectral or prosodic features to deceive a speaker verification system.

There are also weak links from the linguistic or high-level feature aspects. In the text-dependent speaker verification case, it is possible for the attacker to obtain the exact pass-phrase information in advance, while for the text-independent speaker verification case, the attacker can either familiarise the choice of words and speaker style of the target speaker in advance or speak freely, as text-independent speaker verification systems do not have any constraint in the language content for verification.

From the perspective of speaker modelling, most of the systems use a GMM as the basis to model feature distributions. Such an implementation ignores the temporal structure of speech, which also reflects the speaker individuality. On the other hand, voice conversion systems are good at performing frame by frame conversions. In this
way, the loss of temporal structure modelling in speaker verification is a weak link to spoofing attacks. Studies have shown that HMM-based speaker verification systems that capture the temporal structure are more resilient than those without temporal constraint in the face of voice conversion spoofing attacks [155]. But we need to note that the latest voice conversion systems, such as duration embedded HMM [169] and trajectory HMM [8] based systems, are designed to transfer the temporal structure of speech from source to target speaker. Hence, whether temporal modelling techniques can provide some protection to voice conversion spoofing remains an open question.

In view of the fact that short-term spectral features are mostly employed in the state-of-the-art systems, which is known to the technical community [4,125,163], therefore, becoming the weakest link in face of attack. This motivates this thesis to focus the study on voice conversion based on spectral mapping in this chapter.

## 5.5 Experimental protocol

### 5.5.1 Performance evaluation measures

A speaker verification trial where the test and enrolment utterances share the same speaker identity variable is a genuine trial; otherwise, it is called an impostor trial. Given a test sample, the acceptance or rejection decision made by a verification system falls into one of the four groups shown in Table 5.1, where false acceptance and false rejection are the misclassifications. Often classifier parameters are optimised to obtain low equal error rate (EER), corresponding to a verification threshold that results in the false acceptance rate (FAR) and false rejection rate (FRR) being equal.

<table>
<thead>
<tr>
<th>Decision</th>
<th>Acceptance</th>
<th>Rejection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genuine test</td>
<td>Correct acceptance</td>
<td>False rejection</td>
</tr>
<tr>
<td>Impostor test</td>
<td>False acceptance</td>
<td>Correct rejection</td>
</tr>
</tbody>
</table>

In a spoofing attack scenario, a speaker verification system will be unaware of the attack and is deployed with a fixed threshold, assuming that the testing samples are natural human voices. FAR is therefore a natural criterion for evaluating vulnerabilities
of a speaker verification system under spoofing attack. Formally, let $\text{FAR}(\theta, \mathcal{D})$ and $\text{FRR}(\theta, \mathcal{D})$ denote FAR and FRR, evaluated at operating point (threshold) $\theta$ on dataset (corpus) $\mathcal{D}$. Let $\mathcal{D}_{\text{base}}$ denote a baseline corpus consisting of genuine and zero-effort impostor trials (i.e. no dedicated spoofing attempts). Further, let $\mathcal{D}_{\text{attack}}$ be a corpus that shares the same genuine trials as $\mathcal{D}_{\text{base}}$ but in which all the impostor trials have been replaced by voice conversion samples simulating a dedicated attack. With these notations, the protocol is:

1. Determine EER threshold on $\mathcal{D}_{\text{base}}$:

   $$\theta_{\text{EER}} = \arg \min_{\theta} |\text{FRR}(\theta, \mathcal{D}_{\text{base}}) - \text{FAR}(\theta, \mathcal{D}_{\text{base}})|$$

2. Compute $\text{EER}(\theta_{\text{EER}}, \mathcal{D}_{\text{attack}})$ and $\text{FAR}(\theta_{\text{EER}}, \mathcal{D}_{\text{attack}})$ and observe their relative increase w.r.t. baseline dataset.

5.5.2 Baseline and spoofing datasets

In light of the mass market adoption of speaker verification technology in smartphone [123] and in remote/online user access control, this study decides to focus on telephony and mobile device quality speech.

NIST SRE 2006: Telephone quality dataset

For telephone quality speech, we choose the NIST SRE 2006 corpus\(^5\), which is a standard benchmark dataset for text-independent speaker verification. The training utterances for target speaker models and the evaluation trials are directly taken as a subset of the core task, $1\text{conv}4\text{w}$-$1\text{conv}4\text{w}$, from the original corpus. As the average duration of the speech samples in NIST SRE 2006 is five minutes, this poses a computational challenge when performing voice conversion. This study hence uses significantly less evaluation trials compared to the actual NIST SRE 2006 tasks, and use less impostor trials. In particular, the speaker verification task conducted for this study consists of 6,693 gender-matched evaluation trials, including 3,946 genuine and 2,747 impostor trials from 504 target speakers. Table 5.2 presents the statistics of the dataset.

Table 5.2: Statistics of the genuine, impostor and impostor via voice conversion (VC) trials (subset of NIST SRE 2006 core task) for the spoofing attack study.

<table>
<thead>
<tr>
<th></th>
<th>Female</th>
<th>Male</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target speakers</td>
<td>298</td>
<td>206</td>
<td>504</td>
</tr>
<tr>
<td>Genuine trials</td>
<td>2,332</td>
<td>1,614</td>
<td>3,946</td>
</tr>
<tr>
<td>Impostor trials</td>
<td>1,615</td>
<td>1,132</td>
<td>2,747</td>
</tr>
<tr>
<td>Impostor trials via VC</td>
<td>1,615</td>
<td>1,132</td>
<td>2,747</td>
</tr>
</tbody>
</table>

Similar to previous studies [143, 170], the utterances used for training the speaker enrolment models and voice conversion functions are disjoint. This study utilises data from the 3- and 8-conversation training sections, $3\text{conv}4\text{w}$ and $8\text{conv}4\text{w}$, of the NIST SRE 2006 corpus to train the conversion functions. One conversation session (approximate 5 minutes) from each speaker is employed to estimate the conversion function, and the duration of each trial for verification is about 5 minutes.

**RSR2015: Mobile device quality dataset**

For mobile device quality speech, the nine sessions of the first two parts of the RSR2015 database [171] are used. This corpus has been recorded using multiple mobile devices and smartphones over nine recording sessions and this corpus can be used as a standard benchmark database for text-dependent speaker verification system development and evaluation. During the recording, a speaker reads 30 pass-phrases for each session of Part I and 30 short commands for each session of Part II. The average duration of the pass-phrases is 3.2 seconds. Two non-overlapping sets of speakers are defined: a background set including 60 male and 60 female speakers, and an evaluation set of 30 male and 30 female speakers. Speakers from the background set are reserved for training a universal background model (UBM) [129] needed for constructing our classifiers.

For the experiments, each speaker from the evaluation set is used both as a target speaker and as an impostor against other speakers of same gender. Out of the 9 sessions available for each speaker, three sessions are used for enrolment (sessions 1, 4 and 7) while the six remaining sessions are used as test material. Note that enrolment
and test sessions are defined so that the recording device used for verification test is different from the one used during the enrolment. To avoid overlapping between speaker model training and conversion function training, the 30 sentences are further split into two groups. Pass-phrases 1 to 10 are used for speaker verification experiments while sentences 11 to 30 are set aside for training the voice conversion function. Thus, 60 utterances from each speaker are used to produce genuine and impostor trials (10 pass-phrases and 6 sessions). The statistics of the trials are presented in Table 5.3. Given this protocol, it is noted that only the genuine and impostor trials with matched pass-phrase and matched gender are considered. That is, the attacker knows the prompted pass-phrase.

Table 5.3: Statistics of the baseline and spoofing datasets from RSR2015 database (VC=voice conversion).

<table>
<thead>
<tr>
<th></th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target speakers</td>
<td>30</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>Genuine trials</td>
<td>1,796</td>
<td>1,797</td>
<td>3,593</td>
</tr>
<tr>
<td>Impostor trials</td>
<td>51,621</td>
<td>51,853</td>
<td>103,474</td>
</tr>
<tr>
<td>Impostor trials via VC</td>
<td>51,621</td>
<td>51,853</td>
<td>103,474</td>
</tr>
</tbody>
</table>

Voice conversion spoofing datasets

To generate the spoofing attack datasets, the test samples for the impostor trials are passed through voice conversion while the genuine trials are kept untouched. This allows this work to focus solely on the effects of spoofing attack. The spoofing attack datasets are designed by repeating the following three steps for each impostor trial:

- Estimate a conversion function between an impostor and a target genuine speaker’s speech;
- Employ the conversion function to modify each test sample of the impostor;
- Adopt the converted speech sample as a testing sample of the impostor.

In practice, the JD-GMM method introduced in Chapter 2 is used to generate the spoofing dataset, and the number of converted trials are made the same as that of the original impostor trials. The genuine trials and original impostor trials are pooled as
Chapter 5. Vulnerability Evaluation of Speaker Verification Under Voice Conversion Spoofing Attacks

A baseline test, and at the same time the genuine trials and impostor trials via voice conversion are mixed as a spoofing attack test. It is expected to see the decisions of genuine trials remain the same between the baseline and the spoofing test, and an increase of false alarm arising from the converted speech samples. In this way, the performance of speaker verification system with and without spoofing attack is comparable and it is able to examine the spoofing attack effect. The actual numbers of trials are presented in Tables 5.2 and 5.3.

5.6 Experimental results and discussion

The objective of our experiments is to evaluate the vulnerabilities of automatic speaker verification systems under voice conversion spoofing attacks. As the study involves both speaker verification and voice conversion techniques, the research problem can be viewed from two different angles: a) examining the performance of speaker verification systems under spoofing attack; and b) analysing the effectiveness of voice conversion as a spoofing approach. In this work, details of the speaker verification systems, such as feature extraction and speaker modelling techniques, are assumed unknown to an attacker. Table 5.4 summarises the configurations of speaker verification systems with corresponding evaluation datasets. NIST SRE 2006 is only used for text-independent systems study while RSR2015 is only used for text-dependent systems study.

Table 5.4: A summary of text-independent (TI) and text-dependent (TD) speaker verification systems used in this study.

<table>
<thead>
<tr>
<th>Model</th>
<th>UBM</th>
<th>Background data</th>
<th>Scoring</th>
<th>Score normalisation</th>
<th>Model type</th>
<th>Speaker model dim.</th>
<th>Mode</th>
<th>Evaluation dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-UBM</td>
<td>2048</td>
<td>SRE 2004</td>
<td>Likelihood ratio</td>
<td>-</td>
<td>Generative</td>
<td>74,728 (2048 x 36)</td>
<td>TI</td>
<td>SRE 2006</td>
</tr>
<tr>
<td>VQ-UBM</td>
<td>2048</td>
<td>SRE 2004</td>
<td>Likelihood ratio</td>
<td>-</td>
<td>Generative</td>
<td>73,728 (2048 x 36)</td>
<td>TI</td>
<td>SRE 2006</td>
</tr>
<tr>
<td>CLDLS-SVM</td>
<td>n/a</td>
<td>SRE 2004</td>
<td>Linear SVM</td>
<td>-</td>
<td>Discriminative</td>
<td>9,139 (3rd order polyn.)</td>
<td>TI</td>
<td>SRE 2006</td>
</tr>
<tr>
<td>GMM-SVM</td>
<td>512</td>
<td>SRE 2004</td>
<td>Linear SVM</td>
<td>ZT-norm</td>
<td>Discriminative</td>
<td>18,432 (512 x 36)</td>
<td>TI</td>
<td>SRE 2006</td>
</tr>
<tr>
<td>GMM-JFA</td>
<td>512</td>
<td>SRE 2004</td>
<td>Likelihood ratio</td>
<td>ZT-norm</td>
<td>Generative</td>
<td>400 (i-vector dim.)</td>
<td>TI</td>
<td>SRE 2006</td>
</tr>
<tr>
<td>PLDA</td>
<td>512</td>
<td>SRE 2004</td>
<td>Likelihood ratio</td>
<td>-</td>
<td>Generative</td>
<td>400 (i-vector dim.)</td>
<td>TI</td>
<td>SRE 2006</td>
</tr>
<tr>
<td>Fusion system</td>
<td></td>
<td></td>
<td>Logistic regression</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TD-GMM</td>
<td>64</td>
<td>RSR2015</td>
<td>Likelihood ratio</td>
<td>-</td>
<td>Generative</td>
<td>2,304 (64 x 36)</td>
<td>TD</td>
<td>RSR2015</td>
</tr>
<tr>
<td>TD-HMM</td>
<td>64</td>
<td>RSR2015</td>
<td>Likelihood ratio</td>
<td>-</td>
<td>Generative</td>
<td>2,304 (64 x 36)</td>
<td>TD</td>
<td>RSR2015</td>
</tr>
</tbody>
</table>

In this section, three case studies are presented. The first case study evaluates the performance of seven text-independent speaker verification systems including the classical GMM-UBM system and current state-of-the-art PLDA system under the same voice conversion attack, while the second case study examines the vulnerabilities of
two variants of text-dependent systems, and also studies how the number of JD-GMM training utterances affects the outcomes. In the third case study, the second case study is extended by having an inquiry into the relationship between voice conversion performance and the respective effects of spoofing attacks to gain further insights.

5.6.1 Parameterizations

Feature extraction for Speaker verification

All the speaker verification systems use the same acoustic front-end consisting of 12 MFCCs with delta and delta-delta coefficients computed via 27-channel mel-frequency filterbank. RASTA filtering, voice activity detection (VAD) and utterance cepstrum mean-variance normalisation (CMVN) are employed as post-processing. The VAD decisions of test segments are derived from the original baseline datasets.

Feature extraction for voice conversion

For NIST SRE 2006 dataset, a speech signal, which is sampled at 8 kHz, is windowed in a 25 ms window with a 5 ms window shift. 30 dimensional Mel-cepstrum coefficients (MCC) are extracted to represent spectral envelopes utilising Mel-cepstral analysis technique [104], and fundamental frequency (F0) is extracted using the RAPT algorithm [172]. During synthesis, MCC parameters are passed to a Mel-log-spectrum approximation (MLSA) filter [104] to reconstruct a speech signal. In practice, the Speech Signal Processing Toolkit (SPTK) [173] is adopted for signal analysis and reconstruction.

For the RSR2015 dataset, a harmonic plus noise model (HNM) based vocoder\(^6\) is used as implemented in [174]. A different vocoding is employed on RSR2015 due to a higher sampling rate (16 kHz). For both datasets, MCCs are converted using JD-GMM described above while F0 is converted by equalizing the means and variances of source and target speakers in log-scale.

\(^6\)http://aholab.ehu.es/ahocoder/
5.6.2 Case study 1: Attacking text-independent systems

In the first set of experiments, the vulnerabilities of seven text-independent speaker verification systems are evaluated under the same voice conversion spoofing attack. The equal error rates of the seven systems before and after voice conversion attack are presented in Table 5.5. It is expected that the fusion system offers the lowest Baseline EER, while the GMM-UBM the highest. PLDA slightly outperforms GMM-JFA, which operates on GMM supervectors. Since GMM-UBM, VQ-UBM and GLDS-SVM systems do not employ intersession compensation techniques, their EERs are above 7 %, whereas the more advanced GMM-SVM, GMM-JFA and PLDA systems achieve EERs lower than 4 %.

Table 5.5: Equal error rate results of seven different speaker verification systems before and after voice conversion attack.

<table>
<thead>
<tr>
<th>Voice conversion</th>
<th>Equal error rates (EER %)</th>
<th>(relative increment %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (Baseline)</td>
<td>7.62</td>
<td>24.86 (226.25)</td>
</tr>
<tr>
<td>Spoofing</td>
<td>22.39 (198.14)</td>
<td>24.79 (247.20)</td>
</tr>
</tbody>
</table>

All the seven classifiers are seriously affected by the voice conversion attack. Indeed, EERs of the GMM-UBM and the PLDA systems increase, respectively, from 7.62 % to 24.86 % and from 2.82 % to 6.63 %. EERs of the two simplest classifiers — GMM-UBM and VQ-UBM — and the two discriminative classifiers — GLDS-SVM and GMM-SVM — increase by three-fold under voice conversion spoofing attack, while EERs of GMM-JFA and PLDA double. The EER of the fusion system also increases by three-fold.

FAR is a more meaningful evaluation measure of spoofing attack. As Table 5.6 indicates, FARs of GMM-UBM, VQ-UBM and GLDS-SVM increase from ~7.5 % to over 50 %, and FARs of GMM-SVM, GMM-JFA and PLDA increase from 3.66 %, 3.22 % and 2.82 % to 40.42 %, 17.14 % and 18.67 %, respectively. FAR of the fusion system increases from 2.09 % to 21.57 %. Even though the baseline FARs are close to each other, the GMM-JFA system experiences less degradation.

For further analysis, the verification scores of the PLDA system before and after attack are presented in Fig. 5.4. It is clear that spoofing attack shifts the impostor’s score distribution towards genuine score distribution. To sum up our observations, we confirm the vulnerabilities of all the seven text-independent speaker verification systems. Even though current state-of-the-art systems, such as GMM-JFA and PLDA
Table 5.6: False acceptance rate (FAR) results of seven text-independent speaker verification systems before and after voice conversion spoofing attack. FAR is obtained by setting the decision threshold to the EER point on the baseline dataset.

<table>
<thead>
<tr>
<th>Voice conversion</th>
<th>GMM-UBM</th>
<th>VQ-UBM</th>
<th>GLDS-SVM</th>
<th>GMM-SVM</th>
<th>GMM-JFA</th>
<th>PLDA</th>
<th>Fusion system</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (Baseline)</td>
<td>7.62</td>
<td>7.51</td>
<td>7.14</td>
<td>3.66</td>
<td>3.22</td>
<td>2.82</td>
<td>2.09</td>
</tr>
<tr>
<td>Spoofing</td>
<td>55.95</td>
<td>56.13</td>
<td>53.17</td>
<td>40.42</td>
<td>17.14</td>
<td>18.67</td>
<td>21.57</td>
</tr>
</tbody>
</table>

systems, and the fusion system work well on natural human speech, they can still be compromised by a voice conversion attack.

Figure 5.4: Score distributions of PLDA system before and after voice conversion spoofing.

5.6.3 Case study 2: Attacking text-dependent systems

The second set of experiments is designed to evaluate the vulnerabilities of text-dependent speaker verification systems. In the experiments, the number of parallel
training utterances is varied from 2 to 20. In particular, 2, 5, 10 and 20 utterances are used, respectively, to estimate the conversion function. They are labeled as VC-2, VC-5, VC-10 and VC-20 in Table 5.7, and the corresponding number of Gaussian components in JD-GMM is empirically set to 4, 8, 16 and 32, respectively.

Table 5.7: Performance of text-dependent speaker verification systems under the same voice conversion spoofing attack. False acceptance rate (FAR) is obtained by setting the threshold to the equal error rate point on baseline dataset. Assuming the impostor knows the pass-phrases.

<table>
<thead>
<tr>
<th>Voice conversion</th>
<th>EER (%)</th>
<th>FAR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TD-HMM</td>
<td>TD-GMM</td>
</tr>
<tr>
<td></td>
<td>Male</td>
<td>Female</td>
</tr>
<tr>
<td>None (Baseline)</td>
<td>2.92</td>
<td>2.39</td>
</tr>
<tr>
<td>Spoofing (VC-2)</td>
<td>3.90</td>
<td>1.78</td>
</tr>
<tr>
<td>Spoofing (VC-5)</td>
<td>5.07</td>
<td>2.51</td>
</tr>
<tr>
<td>Spoofing (VC-10)</td>
<td>7.04</td>
<td>2.82</td>
</tr>
<tr>
<td>Spoofing (VC-20)</td>
<td>8.30</td>
<td>3.12</td>
</tr>
</tbody>
</table>

First, the EERs of TD-HMM speaker verification systems before and after voice conversion attack are compared. As shown in Table 5.7, before spoofing attack, the EERs of TD-HMM systems are 2.92 % and 2.39 % for male and female, respectively. As a result of spoofing attack using 2 utterances for training, the EER of male speakers increases to 3.90 %, however, the EER of female speakers decreases to 1.78 %. This might be because of too few voice conversion training utterances. Indeed, when the number of parallel training utterances is increased, the EERs increase to 8.30 % and 3.12 % for male and female, respectively. When more than 5 utterances are used to estimate the conversion function, the EERs after spoofing attack are higher than that before spoofing attack for both male and female.

The performance of TD-GMM speaker verification systems is then evaluated. Before spoofing attack, the EERs are 4.01 % and 3.67 % for male and female, respectively, which are slightly higher than that of TD-HMM systems. As presented in Table 5.7, even when only 2 utterances are used for estimating the voice conversion function, EERs increase over baseline for both genders. When using 20 utterances to estimate a conversion function, the EERs increase to 13.34 % and 7.31 % for male and female, respectively. It is noted that for both TD-HMM and TD-GMM systems, male speakers have higher EERs than those of female speakers.

The FAR results of both TD-HMM and TD-GMM are also presented in Table 5.7. When using 20 utterances to train a conversion function, the FARs of the TD-HMM
verification systems increase from 2.93 % and 2.39 % of baseline to 21.87 % and 4.68 % after spoofing attack for male and female, respectively. The FARs of the TD-GMM systems increase from 4.01 % and 3.67 % before spoofing attack to 33.23 % and 13.20 % after spoofing attack for male and female, respectively. A similar effect for EERs is observed as the number of training utterances varies.

To sum up, increasing the number of training utterances for voice conversion increases both EERs and FARs. Voice conversion with enough training training data is hence able to move an impostor's feature distribution towards that of a target speaker, and presents an increased threat to both TD-HMM and TD-GMM verification systems. Since TD-HMM uses hidden Markov model to capture both feature distribution and temporal sequence information, it outperforms TD-GMM system which only models the feature distribution even under voice conversion attack. Note that the temporal sequence information remains the same as in the original impostor samples after voice conversion.

Figure 5.5: Score distributions of the male TD-HMM system before and after spoofing.
As a further analysis, the score distributions of the TD-HMM system for males and females are presented in Fig. 5.5 and 5.6, respectively. A similar score shifting pattern for TD-GMM is observed. Trials on the right hand side of the decision threshold are falsely accepted while those on the left hand side are correctly decided as rejected. For male speakers, all the four cases of spoofing attacks consistently move the imposture scores towards the right side of the decision threshold. As for female speakers, there is a similar observation except the case of two utterances (VC-2) which slightly shifts the score distribution towards the left with a reduced score variance. Using more VC training data (VC-5, VC-10 and VC-20), score distribution translates to the right, consistent with the FAR results in Table 5.7. We note that VC-2 shows unsuccessful attack when the threshold is set to EER point, but it might be effective for other settings of the decision threshold. For instance, a system optimised to have a lower false rejection rate (FRR).
Generally, when enough training data is used to estimate the conversion function, voice conversion spoofing attack is able to compromise both TD-HMM and TD-GMM verification systems. This confirms the risk of voice conversion spoofing attack and the vulnerabilities of both TD-HMM and TD-GMM verification systems.

5.6.4 Case study 3: Voice conversion performance vs spoofing effect

In this section, we attempt to explain the results from the perspective of voice conversion and establish the relationship between the voice conversion performance and that of speaker verification. From the previous text-dependent analysis, we have the following observations: a) the female systems have lower EERs and FARs than the male systems, b) it shows unsuccessful spoofing attack simulated by voice conversion using only two utterances on the female TD-HMM system, c) the more training data for voice conversion, the higher EERs and FARs of both TD-HMM and TD-GMM under spoofing attacks.

In voice conversion, spectral distortion, in particular Mel-cepstral distortion (MCD) [34], is frequently used as an objective evaluation measure to predict voice conversion performance. MCD is calculated between the source or converted speech and the reference target speech to measure the distance between two speech signals, as follows

$$\text{MCD}[\text{dB}] = 10 \frac{\ln 10}{\ln 10} \sqrt{2 \sum_{d=1}^{D} (c_d - c'_d)^2}, \quad (5.1)$$

where $D$ is the dimension of MCC feature, $c_d$ is the $d$-th dimension reference target feature, and $c'_d$ is the $d$-th dimension source feature with or without conversion. A lower MCD value indicates higher similarity of the compared speech signals. From the perspective of EER and FAR, a lower MCD value may indicate higher EER and FAR, as higher similarity between two speech signals implies a more difficult classification task. To calculate spectral distortion, we randomly select 5,000 source-target utterance pairs for each gender. It is noted that RSR2015 pass-phrase part is a parallel dataset, that is, each speaker speaks the same words. Dynamic time warping (DTW) is used to perform optimal frame alignment between source and target utterances to get the frame
pairs for calculating MCD. The converted utterance shares the alignment information with source utterances. Hence, the spectral distortion of a source utterance with and without conversion to a target utterance is comparable. The calculation is done frame-by-frame and we report the average distortion.

Fig. 5.7 (a-c) presents the comparison of between spectral distortions and FARs. Without voice conversion, the spectral distortions between source and target speech are 7.81 dB and 8.07 dB for male and female, respectively. Higher spectral distortion of female implies larger variability across speakers, therefore, it is easier to classify female speakers and is more difficult to estimate the conversion function for female speaker conversion. Refer to Table 5.7, the female TD-HMM and TD-GMM systems have lower EERs than that of the male systems.

When increasing the number of training utterances, spectral distortions decrease from 6.86 dB and 7.09 dB of two utterances to 6.46 dB and 6.79 dB of 20 utterances for male and female, respectively. Instead, FARs of both TD-HMM and TD-GMM increase as expected. This changing trends between spectral distortions and FARs go opposite directions. It is also noted that for each conversion case, female speakers always have higher spectral distortion than that of male speakers. This phenomenon is observed in the EER and FAR results, where female speakers always achieve lower EERs and FARs than those of male speakers.

5.7 Conclusions

This study examined the vulnerabilities of text-independent and text-dependent speaker verification systems under voice conversion attack using two benchmark datasets, the same protocol and the same voice conversion method. The experimental results confirmed the vulnerabilities of text-independent and text-dependent systems. The main findings are:

1. Speaker verification systems, including the state-of-the-art i-vector approach, were found highly vulnerable to voice conversion attacks.

2. HMM-based text-dependent systems, utilising time sequence information, were found more resistant in face of spoofing in comparison to systems lacking temporal modelling.
Figure 5.7: Comparison across spectral distortions, FARs of TD-HMM systems and that of TD-GMM systems on the RSR2015 dataset. (a): FARs of male and female TD-HMM systems. No conversion means before spoofing attack. VC-2, VC-5, VC-10 and VC-20 represent different voice conversion attacks. (b): FARs of male and female TD-GMM systems. (c): Spectral distortions of male and female voice conversion systems.

3. Comparing speaker classifiers of different complexity, the more advanced techniques such as JFA and PLDA were found more resistant against attacks in comparison to simpler methods without inter-session compensation.

4. Successful attacks to text-dependent classifiers require sufficiently many training utterances – in the findings, five or more for the set-up considered.

The first finding independently confirms the previous findings by other research groups involving voice conversion attacks (e.g. [146–150]). Despite the rapid progress in base technologies for speaker verification, our new evidence with the recent classifiers –
JFA and PLDA – clearly indicates that the threat is tangible even with modern techniques. The second finding is expected and even often claimed without further evidence. Through a set of controlled experiments involving both text-independent and two types of text-dependent systems, this study has quantitatively shown that temporal information is helpful against voice conversion attacks.

The third finding is the least obvious to the authors. In fact, since JFA and PLDA involve explicit inter-session (intra-speaker) compensation, one might have expected that a spoofed test sample will be considered as belonging to the natural intra-speaker variation space of the target, producing large verification scores. The fact that PLDA was degraded less than the old classifiers could be simply because it is a generally more robust classifier with inclusion of utterance constraints defined by the hyper-parameters of the i-vector extractor and PLDA. As the hyper-parameters are trained from natural speech, the test samples produced by vocoding techniques will match poorly training set characteristics.

The fourth finding is naturally expected, and one of the novelties of this study was to establish a link between voice conversion quality (Mel-cepstral distortions) and spoofing success (false acceptance rates). In the current experiments, this was observed for female speakers with TD-HMM when only two voice conversion training utterances were used (VC-2), but the effect disappears with increased number of VC training utterances. With too few VC training utterances, the speaker transformation (JD-GMM) might be undertrained, causing conversion artefacts that the TD-HMM correctly considers being far off from natural speech, leading improved separation of genuine and (converted) impostor score distributions.

A perfect voice conversion would produce an exact copy of the target speaker’s test utterance (MCD of 0). By construction, the spoofed and genuine scores would completely overlap. But since the converted speech scores we have observed (e.g. Fig. 5.4) are still far away from the genuine scores, the conversion process is imperfect. This could be either due to vocoder artefacts or limitations of the JD-GMM conversion. JD-GMM was chosen due to reasons of popularity and simplicity without considering more advanced spectral or prosody conversion techniques. In general, it is expected that more advanced voice conversion techniques yield higher relative degradations, but this was seen unnecessary given that even the simplest technique leads to unacceptable error rates. It would be interesting to examine the effectiveness of more advance
voice conversion techniques and alternative features, such as prosody, both at speaker verification and voice conversion sides.

The confirmation of the vulnerabilities of speaker verification systems under voice conversion spoofing attacks suggests urgent development of countermeasures to protect speaker verification systems from dedicated spoofing. There are two possible directions to enhance the performance under spoofing: (1) implementing stand-alone countermeasures as a complementary component to speaker verification systems. Unlike a human ear, most classifiers will accept even nonsensical, artificially produced input speech when constructed to match target speech features [154]. The second direction is to (2) improve the fundamentals of speaker verification, such as including time sequence information and high level features. Likely, both directions will play an important role in near future and are left as a follow-up work.
Chapter 6

Conclusions and Future Work

This thesis has focused on the spectral mapping for voice conversion and its application. It has been shown that it is a challenge task to produce high quality speech which sounds like a specific target speaker when training data are limited. Currently, most voice conversion systems suffer the over-smoothing or over-fitting problems in such a training scenario. To address these issues, this thesis has proposed two novel approaches, inspired by the research from speaker recognition and speech enhancement. One of the major tasks in speaker recognition is to deal with speaker individuality, which is also a key task in voice conversion. Similarly, techniques for speech enhancement aim to map noise speech signal to obtain the corresponding clean version without changing the language content, and hence have similar objectives to voice conversion. Lastly, the malicious use of voice conversion technology attacking speaker verification is examined to assess the vulnerability of current state-of-the-art speaker verification systems. The reminding of this chapter is organised as follows: Section 6.1 summaries the contributions in this thesis; and Section 6.2 discusses the possible future research directions that might be explored.

6.1 Contributions

The implementation of a robust conversion function is one of the most important modules of a voice conversion system. This thesis have contributed two novel techniques to improve the realisation of conversion functions under sparse training data condition.
(first contribution) and for high-dimensional spectral feature modelling (second contribution). In addition, The thesis also conducted a systematic study examining the vulnerability of nine state-of-the-art speaker verification systems to voice conversion attacks.

### 6.1.1 Mixture of factor analyzers using nonparallel data

The first contribution of this thesis proposes to use nonparallel data from non-target speakers to improve the modelling of the voice conversion function using a mixture of factor analysers model. This work has been published in the IEEE Signal Processing Letters [19]. This idea follows the strategy used by joint factor analysis in speaker verification, which decomposes a speech vector into three independent components, namely speaker-independent, speaker-dependent and channel-dependent components. The speaker-independent component is related to the linguistic information contained in the speech signal; the speaker-dependent component describes the speaker individuality, and the channel-dependent component represents the transmission channel of the signal. A similar decomposition was applied to the clean voice conversion data, to separate a speech vector into speaker-independent phonetic and speaker-dependent speaker-specific components. The speaker-dependent component was represented by a low-dimensional set of variables via factor loadings. In this way, the nonparallel datasets from other speakers were used to estimate the speaker-independent phonetic component and the factor loadings. The parallel data from the source and target speakers were used to estimate the low-dimensional speaker-vector to represent the source and target speakers’ characteristics.

Experiments were conducted on the CMU ARCTIC corpus. It is observed that when there is a limited amount of training data, for example, 2 to 5 seconds speech, the proposed mixture of factor analysers (MFA) method works well comparing with the classic JD-GMM method. In particular, the spectral distortion of the proposed MFA method is 0.2 dB lower than the JD-GMM method. In the subjective evaluation, both speech quality and speaker similarity of the proposed MFA are significant higher than that of the JD-GMM method. However, if sufficient training data are available, the proposed mixture of factor analysers method does not work as well as the JD-GMM methods, as there are more parameters in the JD-GMM method than the proposed
method. This is the limitation of the proposed method.

### 6.1.2 Exemplar-based sparse representation with residual compensation

With the success of mixture of factor analysers method using priors on very small dataset, the second contribution of the thesis is an exemplar-based sparse representation with residual compensation method for voice conversion. This work has been published in the 8th ISCA Speech Synthesis Workshop (SSW8) [20] and its extended version is published in the IEEE/ACM Transactions on Audio, Speech and Language Processing [21].

Following the strategy of exemplar-based sparse representation for source separation, this thesis used a similar idea to model the target speech from real speech segments, called exemplars. In this way, high-resolution features with spectral details were directly used without dimensionality reduction, and the statistical averaging phenomenon in the statistical parametric approaches were reduced by introducing sparsity constraint. In the proposed framework, multiple-frame exemplars were first used to include temporal information constraint. Then, spectral compression was applied to the spectra to balance the low-density but important frequencies. Finally, residual compensation was applied to compense the modelling errors in the sparse representation.

Experiments were conducted on the VOICES corpus to examine the proposed method. From both the objective and subjective evaluations, it has been shown that the proposed method is able to achieve similar performance to current state-of-the-art voice conversion methods, and is more flexible and robust than existing methods. In addition, the proposed framework is complementary with the conventional parametric approaches, in the sense that the conventional parametric approaches can be adopted for residual compensation.

There are also limitations of the proposed exemplar-based sparse representation with residual compensation method. The major drawback is the computational cost and memory footprint. The proposed method is considerably higher in computational cost and memory footprint than the state-of-the-art voice conversion methods. This drawback may be avoided by using smaller dictionary through dictionary learning or
6.1.3 Voice conversion spoofing attacks to speaker verification systems

The third contribution of this thesis examines the use of voice conversion technology as a spoofing attack approach to examine the vulnerability of speaker verification systems. The task of speaker verification technology is to automatically accept or reject a claimed identity based on a speech sample. On the other hand, voice conversion can change the speaker identity of an impostor to mimic the target genuine speaker. Thus, the security of speaker verification systems under voice conversion spoofing attack becomes a major concern when deploying speaker verification systems. This work has been published in the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2012 [22], Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC) 2012 and 2013 [6,23], and Interspeech 2013 [24].

In Chapter 5, the vulnerability of text-independent speaker verification systems was first analysed. The experimental results reported on the NIST speaker recognition evaluation (SRE) 2006 dataset confirmed the vulnerability of various speaker verification systems, from the classic GMM-UBM system to the current state-of-the-art i-vector PLDA system, to voice conversion spoofing. Then, the vulnerability of text-dependent speaker verification systems was assessed. Similar to text-independent systems, text-dependent speaker verification systems can be easily fooled by a simple voice conversion system.

Finally, the spoofing performance was linked with the voice conversion performance in terms of spectral distortion. In general, speaker verification systems model the feature distribution of feature representations while voice conversion technology is to shift the source speaker’s feature distribution towards that of the target speaker. This explains why the speaker verification systems can be overcome by voice conversion technology.
6.2 Future Directions

The objective of the thesis has been to develop novel algorithms to improve voice conversion function under limited training data condition. This has resulted in the development of several techniques that address the issues in voice conversion. However, the algorithms were developed with some specific constraints, such as parallel training data. To generalise the proposed algorithms, some issues need to be solved in the future work. In addition, with the confirmation of vulnerability of speaker verification under voice conversion spoofing attacks, a significant amount of work is required to prevent such a spoofing attack and to enhance the security of speaker verification systems. To this end, some future research directions are proposed.

**Nonparallel training:** Currently, this thesis assumes that parallel data are available to train the conversion function. However, in some applications, such as cross-lingual voice conversion, it is impossible to collect such a parallel dataset. It is hence interesting to find ways to perform nonparallel training for voice conversion. Such nonparallel training techniques may focus on new frame alignment techniques or using model adaptation techniques similar to that used in statistical parametric speech synthesis.

**Dictionary learning:** In the proposed exemplar-based sparse representation framework, the computational cost is considerably high, as a large set of exemplars are used in the dictionary. It may be possible to perform dictionary selection or dictionary learning to find a small size of dictionary for voice conversion. The small-size dictionary will reduce the computational cost and memory footprint considerably.

**Frequency warping:** The proposed sparse representation with residual compensation technique could be integrated with the frequency warping based voice conversion techniques. In the literature, it is reported that frequency warping based voice conversion techniques could achieve higher speech quality than the statistical parametric methods [88]. However, in existing implementation of frequency warping function [55], GMM is used to produce weighted warping function, and hence is unable to benefit from long-term temporal information. The proposed sparse representation method could estimate activation weights to interpolate the
warping functions to replace the posterior from GMM. In addition, the residual compensation idea could be applied to the warped spectra to compensate the amplitude different between the warped and the target spectra.

**Anti-spoofing:** In the spoofing attack study, it has been demonstrated that existing state-of-the-art speaker verification systems can be easily overcome by simple voice conversion technique. It confirms the vulnerability of current speaker verification systems and highlights the risk of deploying of such systems without countermeasures. It is hence important and necessary to develop countermeasures for anti-spoofing attacks. One of the directions could be to create a stand-alone synthetic speech detector to capture the artefacts of synthetic speech, and use such a detector to distinguish natural and synthetic speech.
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


